

Ontology Based Grid Information Interoperation

Hai Jin, Sheng Di, Shengli Li, Ling Chen, Li Qi, Chengwei Wang

Cluster and Grid Computing Lab

Huazhong University of Science and Technology, Wuhan, 430074, China

hjin@hust.edu.cn

Abstract

A series of problems arises with the emergence and fast development of grid. Among all the problems, one of them is how to smoothly connect heterogeneous grid platforms. There are a number of ways to solve this problem and the relatively more efficient one is using ontology. Based on this method, we focus on alleviating semantic inconsistency of various platforms and design a novel ontology. This ontology is depicted by Resource Description Framework (RDF) and transacted by adapters. Finally, we test our method and analyze its performance.

1. Introduction

Grid computing enables coordinated resource sharing and problem solving in dynamic, multi-institutional organizations [1]. With the revolutionary impact of large-scale resource sharing and virtualization within both science and industry, grid has become a trend of distributed computing and it has come into widespread use.

Numerous organizations, corporations and universities plunged into the grid field and developed their own grid platforms. At present, there have been quite a few related studies, such as UNICORE [2], CERN Data Grid [3], Globus [4], JaWS [5] and ChinaGrid [6]. However, there are no common uniform standards or specifications for them. Thus, the interoperation among different platforms has become a huge challenge to the development of grid. With respect to this issue, semantic technology [7] provides demanders with a common understanding of the requested information, and one of the most popular and most efficient methodologies is using ontology.

What is ontology? The most common definition is “a formal, explicit specification of a shared conceptualization” [8]. “A conceptualization”, in this context, refers to an abstract model of how people regard objects in the world. “An explicit specification” means that the concepts (including entities and relationships) in any ontology should be defined

clearly and explicitly. “Shared” implies that the main purpose of any ontology is generally to be used and reused across different applications and communities.

There are mainly two advantages of ontology:

1) Ontology can be used to represent and deduce knowledge. Ontology provides a vigorous backbone for knowledge presentation and inference, and also focuses on knowledge sharing.

2) Because of its high ability of expression, ontology can also be adopted for interoperation. In fact, the promise of ontology is “a shared and common understanding of a domain that can be communicated between people and application systems” [9]. Therefore, an efficient way to implement interoperation is designing an appropriate ontology and developing the corresponding adapters.

Ontology has 4 scenarios: Neutral Authoring, Ontology as Specification, Common Access to Information, and Ontology-based Search [8]. We mainly adopt the third one, Common Access to Information, to solve our problem. The goal of this scenario is not only to provide ontology, a knowledge backbone, but design a kind of adapter for grid platforms to be connected. So, based on this scenario, grid platforms could be designed in a freedom as large as possible. In this paper, we focus on alleviating semantic inconsistency of different grids and designing a novel ontology and a kind of adapter for interoperation between various grid platforms. In other words, we design an interoperation mechanism to connect heterogeneous grid platforms.

The rest of this paper is organized as follows: Section 2 describes related work. In Section 3, we present the Grid Information Ontology. We give detailed explanations about grid information interoperation based on ontology in Section 4. Section 5 discusses our implementation method and analyzes its testing results. Finally, we conclude and discuss future work.

2. Related Work

There are several related works to solve the issue about interoperation. We list and compare them as follows.

- ProActive [10]: ProActive is a platform used to provide a solution to the problem of software reuse, integration and deployment for parallel and distributed computing (including grid computing, mobile and ubiquitous distributed computing on the Internet, etc.).
- Combining FT-MPI with H2O [11]: This work presents a kind of framework used to bind FT-MPI and H2O technology. It provides a fault-tolerance MPI Environment and owns the advantages of H2O.
- WSDM [12]: WSDM (Web Service for Distributed Management) is a kind of specification for interoperation between various platforms based on Web Service. Its biggest advantage is allowing administrators defining the relationships between different terms in terms of a common standard.

Comparing with all these related works, we combine registry concept with ontology. This idea makes Our Grid Information Interoperation Platform be able to smoothly connect different grid platforms in an extent, especially from the aspect of semantic heterogeneity [13].

3. Grid Information Ontology

Grid Information is a pivotal part in grid field. As to the information, the most important objects are widely distributed and shared resources. One of the common and efficient ways to manage these resources is designing an Information Center, or Registry. Compared with it, we design a kind of ontology (the backbone of our interoperation platform) to represent resources and their relationships between each other.

We define ontology as a graph with not only entities and edges but rules. The related definitions are listed as follows:

- **Definition 1. Entity:** a class of resources that can be differentiated. It can also be named class. It is marked as “v” in this paper.
- **Definition 2. Property:** a restrictive resource that belongs to some entity: v. It is marked as “v.p” in this paper.
- **Definition 3. Entity Set:** $V=\{v \mid v \text{ is an entity in ontology}\}$
- **Definition 4. Property Set:** $P=\{p \mid v.p, v \in V\}$
- **Definition 5. Relationship Rule:** $R=\text{some kind of rule based on semantics} = \{r_1, r_2, \dots, r_n\}, r_i$

refers to a rule, such as registry manages domains.

- **Definition 6. Edge:** $e= \{n_1, \alpha, n_2\}, n_1 \in V, n_2 \in V \cup P, \alpha \in R$
- **Definition 7. Edge Set:** $E=\{e \mid e = \{n_1, \alpha, n_2\}, n_1 \in V, n_2 \in V \cup P, \alpha \in R\}$
- **Definition 8. Ontology:** $O=(G, R)=(V, E)$

Definition 8 reveals a knowledge-architecture with objects (Entities and Properties) and the relationships between them. At present, popular objects studied in grid mainly include Web Services, applications, transmitted data, work flow, and other resources. Hence, the key work to build a Grid Information Ontology is delving into their deep meanings and relationships. Our Grid Information Ontology Schema is shown in Figure 1 (depicted via RDF).

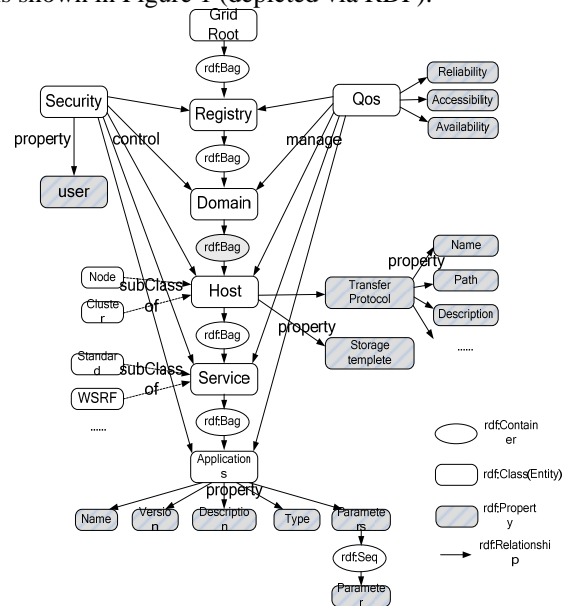


Figure 1. Grid Information Ontology Schema

In this schema, Grid Root is the root object. It owns one or more registries. Each registry manages several domains (Domain is a concept similar to virtual organization in grid [14]) and each domain can have a number of hosts. Any deployed service belongs to some specific host, and each application belongs to a service. The relationship between domain and host is different to others because it is permitted that one host belongs to more than one domains. In addition, Security and Qos are two administrative entities. Any entity in the ontology owns several properties to limit and describe it. Figure 1 displays some of all the properties. Figure 2 shows a small portion (rdf:Bag contains relationship between domain and host) of the ontology model.

```

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:d="
http://grid.hust.edu.cn/domain/contain#">
  <rdf:Description rdf:about="http://grid.hust.edu.cn/domain/001">
    <d:hosts>
      <rdf:Bag>
        <rdf:li rdf:resource="http://grid.hust.edu.cn/hosts/hosts101"/>
        <rdf:li rdf:resource="http://grid.hust.edu.cn/hosts/hosts102"/>
        <rdf:li rdf:resource="http://grid.hust.edu.cn/hosts/hosts104"/>
        <rdf:li rdf:resource="http://grid.hust.edu.cn/hosts/hosts106"/>
      </rdf:Bag>
    </d:hosts>
  </rdf:Description>
  <rdf:Description rdf:about="http://grid.hust.edu.cn/domain/002">
    <d:hosts>
      <rdf:Bag>
        <rdf:li rdf:resource="http://grid.hust.edu.cn/hosts/hosts101"/>
        <rdf:li rdf:resource="http://grid.hust.edu.cn/hosts/hosts102"/>
        <rdf:li rdf:resource="http://grid.hust.edu.cn/hosts/hosts103"/>
        <rdf:li rdf:resource="http://grid.hust.edu.cn/hosts/hosts201"/>
        <rdf:li rdf:resource="http://grid.hust.edu.cn/hosts/hosts202"/>
        <rdf:li rdf:resource="http://grid.hust.edu.cn/hosts/hosts203"/>
      </rdf:Bag>
    </d:hosts>
  </rdf:Description>
  <rdf:Description rdf:about="http://grid.hust.edu.cn/domain/003">
    <d:hosts>
      <rdf:Bag>
        <rdf:li rdf:resource="http://grid.hust.edu.cn/hosts/hosts301"/>
        <rdf:li rdf:resource="http://grid.hust.edu.cn/hosts/hosts302"/>
        <rdf:li rdf:resource="http://grid.hust.edu.cn/hosts/hosts304"/>
        <rdf:li rdf:resource="http://grid.hust.edu.cn/hosts/hosts201"/>
        <rdf:li rdf:resource="http://grid.hust.edu.cn/hosts/hosts102"/>
      </rdf:Bag>
    </d:hosts>
  </rdf:Description>
</rdf:RDF>

```

Figure 2. A Small Portion of Ontology Model

We can do some inferences according to the relationships in the ontology schema. These inferences can be used at the situations in which some entities are ambiguous. Formula (1) and Formula (2) are two examples.

$$\begin{aligned}
 & \text{Application1.host=Host1} \\
 & \rightarrow (\text{Application1, containedBy, Host1}) \quad (1) \\
 & \quad (\text{Host1, containedBy, Service1}), \\
 & \quad (\text{Service1, containedBy, Domain1}) \quad (2) \\
 & \rightarrow (\text{Host1, containedBy, Domain1})
 \end{aligned}$$

4. Interoperation Based on Ontology

The ontology is a uniform backbone for grid information. It is a common specification used to coordinate different grid architectures. We adopt a flexible ontology scenario, Common Access to Information, to organize our interoperation platform. Figure 3 shows this intercommunicated frame.

Operational resource Pool in the frame refers to a pool containing all managed objects, such as entities, properties. They conform to the common ontology. Adapters are key modules that take charge of mapping and translating the semantic terms across platforms. We design a set of translator algorithms in each adapter to execute the fusion process.

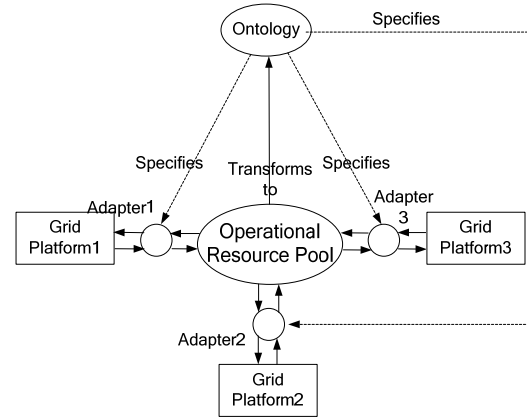


Figure 3. Common Access to Information Frame

There are three core valuables used in the algorithm listed as follows:

schema_mapping_list: This list records the mappings of terms between some grid platform and the standard ontology schema. It is submitted by some platform's administrator.

infer(term): Deduce the mapping of some term in ontology with rules, such as formula (1), formula (2).

exists(lack key entity): It returns true when lacking some key entity. ("key entity" means this entity can not be omitted in Grid Information Ontology, for example, host is a key entity to service)

The algorithm shown in Figure 4 is used to translate the semantics from some grid platform to the standard operational resources.

The algorithm shown in Figure 5 is used to translate the semantics from the standard operational resources to some grid platform.

The algorithm, check (schema_mapping_list), is used to check up the integrality of the schema_mapping_list submitted by some platform's administrator. The schema_mapping_list records mappings between the terms of ontology schema and those of some other platform. If the list lacks an indispensable mapping of some term, the adapter that executes this algorithm will do a series of approaches to repair it. This algorithm is shown in Figure 6.

Hence, the whole process of interoperating grid information between different grid platforms is:

- 1) The administrators of the two target grid platforms respectively submit their own information schemas according to some prescriptive format.
- 2) The corresponding adaptors analyze the submitted schemas and check up their integralities.
- 3) The adaptors fuse the semantic discrepancies between different grid platforms by transforming them to common operational resources and listen to request events.

```

Input: schema_mapping_list, Registry's recorded information
          (XML Streams/Files)
Output: ontology or failing report
1. begin
2.   get the schema_mapping_list and Registry's recorded information;
3.   call check-up-integrity function: check (schema_mapping_list);
4.   if (check(schema_mapping_list)=true) then
5.     begin
6.       construct ontology;
7.       construct the corresponding resources with schema_mapping_list;
8.       construct Registry's recorded information;
9.     end
10.  else // check(schema_mapping_list)=false
11.    begin
12.      report the failure of check-up-integrity;
13.    end
14. end

```

Figure 4. Grid Platform→Operational Resources

```

Input: some ontology and mapping_list
Output: XML files according to the schema of some target grid platform's
Registry
1. begin
2.   get the schema_mapping_list;
3.   get Registry's recorded information;
4.   call check (schema_mapping_list)//check-up-integrity function
5.   if (check(schema_mapping_list)=true) then
6.     begin
7.       create XML files with ontology and schema_mapping_list;
8.     end
9.   else // check(schema_mapping_list)=false
10.  begin
11.    report the failure of check-up-integrity;
12.  end
13. end

```

Figure 5. Operational Resources→Grid Platform

```

Input: schema_mapping_list
Output: true or false
1. begin
2.   foreach(terms in ontology schema) do
3.     begin
4.       get one term: t;
5.       if(t has not been mapped in schema_mapping_list) then
6.         begin
7.           if (inferable(term)=true) then // the term is inferable
8.             begin
9.               infer(term);
10.              add the mapping to schema_mapping_list;
11.            end
12.          else if (exists(lack key entity)=true) then
13.            begin
14.              return false;
15.            end
16.          else
17.            begin
18.              set a default value to the term;
19.            end
20.          end
21.        end
22.      begin
23.        return true;
24.      end
25.    end
26.  return true;
27. end

```

Figure 6. check-up-integrity function: check(schema_mapping_list)

5. Implementation and Performance

We test our grid-information-interoperation platform with two grid platforms, CGSP (ChinaGrid Support Platform) [15] and GPE (UNICORE) [16].

5.1 Preliminary Implementation

As mentioned above, the pivotal work of interoperation is mapping semantics by a set of adapters. We give some sampling mappings between the information schemas of grid platforms and our ontology schema in Figure 7. Based on these mappings, we made a platform and a console to connect two different grid platforms.

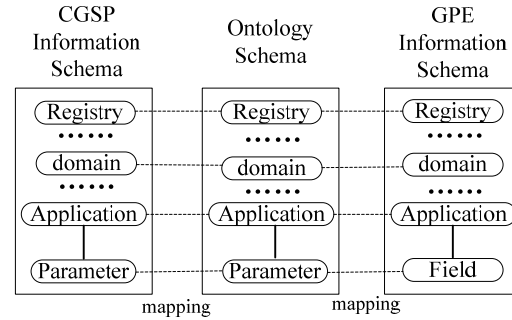


Figure 7. Mapping between Grid Platform Information Schemas and Ontology Schema

5.2 Evaluation of Performance

The performance of our Grid Information Interoperation Platform mainly lies on the number of resources (including entities and properties) to be created in translating semantics. However, properties are lightweight resources because their information is simple, so we just focus on the impact to the performance by creating or visiting entities. According to practical instances, we assume that the number of Registry and Domain is constant. In the practical implementation, we limit the number of services and applications to a scale to keep the whole performance. Table 2 shows the details.

Table 1. The Precondition of The Information of The Target Grid Platform

Entity	Number	Cost	Range
Registry	n_1	c_1	Constant
Domain	n_2	c_2	Constant
Host	n_3	c_3	Variable
Service	n_4	c_4	$[0, n_{4max}]$
Application	n_5	c_5	$[0, n_{5max}]$

In the practical testing, we assume $n_1=5$, $n_2=10$, $n_{4max}=10$, $n_{5max}=20$, and we get the testing results shown in Figure 8.

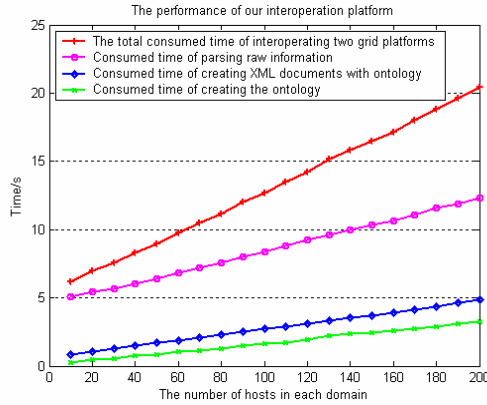


Figure 8. The Performance of Our Interoperation Platform

$$O(n) = \sum_{i=1}^5 (c_i \cdot \prod_{j=1}^i n_j) \quad (3)$$

We deduce the formula (4) from the formula (3).

$$O(n) = K_1 \cdot n_3 + K_2$$

$$(K_1 = n_1 \cdot n_2 \cdot c_3 + n_1 \cdot n_2 \cdot n_{4max} \cdot c_4 + n_1 \cdot n_2 \cdot n_{4max} \cdot n_{5max} \cdot c_5) \quad (4)$$

$$K_2 = n_1 \cdot c_1 + n_1 \cdot n_2 \cdot c_2)$$

From Formula (4) and Figure 8, we notice that the K_1 is the key coefficient impacting the performance of creating ontology. Although the consumed time increases linearly, it is much shorter than that of other steps, such as the time of parsing information and that of creating output documents (XML) from the ontology. At any rate, the whole performance is acceptable for interoperation.

6. Conclusion and Future Work

Interoperation is a big issue for connecting two heterogeneous grid platforms. Our main contribution is designing a uniform grid information view, Grid Information Ontology, and the corresponding translator algorithm to fuse the semantic inconsistencies between grid platforms. The algorithm is programmed into a kind adapter that provides interfaces for target platforms' administrators. We test this method with two grid platforms, CGSP and GPE, and analyze the testing performance.

Our future work should focus on how to improve the performance of our interoperation platform. We will study some methods to optimize our design and algorithms.

References

- [1] I. Foster, C. Kesselman, and S. Tuecke, "The anatomy of the Grid: Enabling scalable virtual organization", *International Journal of Supercomputer Applications*, 15(3), 2001, pp.200-222.
- [2] Unicore project, <http://www.unicore.org/>
- [3] CERN Data Grid project, <http://cern.ch/eu-datagrid>
- [4] Globus project, <http://www.globus.org/>
- [5] S. Lalis and A. Karipidis, "JaWS: an open market-based framework for distributed computing over the Internet", *Proceeding of the 1st International Workshop on Grid Computing (GRID)*, 2000, pp.36-46.
- [6] ChinaGrid project, <http://www.chinagrid.edu.cn/>
- [7] T. Gruber, "A translation approach to portable ontologies", *Knowledge Acquisition*, 1993, pp.199-220.
- [8] M. Uschold and M. Gruninger, "Ontologies and semantics for seamless connectivity", *SIGMOD Record*, 33(4), 2004, pp.58-64.
- [9] D. Fensel, "Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce", Springer-Verlag, 2000.
- [10] L. Baduel, F. Baude, D. Caromel, A. Contes, F. Huet, M. Morel, and R. Quilici, "ProActive: Programming, Composing, Deploying for the Grid", *GridUse*, 2004, pp.12-39
- [11] D. Kurzyniec and V. Sunderam, "Combining FT-MPI with H2O: Fault-Tolerant MPI Across Administrative Boundaries", *International Parallel and Distributed Processing Symposium (IPDPS)*, 2005, pp.120-129
- [12] WSDM Specification: <http://www.oasis-open.org/committees/wsdm>
- [13] J. M. Bradshaw, G. Boy, E. Durfee, M. Gruninger, H. Hexmoor, N. Suri, M. Tambe, M. Uschold, and J. Vitek, "Software Agents for the Warfighter", *ITAC Consortium Report*, AAAI Press/The MIT Press, Cambridge, Massachusetts, 2002.
- [14] I. Foster and C. Kesselman, *The Grid Blueprint for a New Computing Infrastructure* (Second Edition), Morgan Kaufmann, 2004.
- [15] CGSP project, <http://www.chinagrid.edu.cn/CGSP/>
- [16] GPE project, <http://gpe4gtk.sourceforge.net>