Ontology Based Grid Information Interoperation*

Sheng Di, Hai Jin, Shengli Li, Ling Chen, Li Qi, Chengwei Wang
Services Computing Technology and System Lab
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 designing 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 DataGrid [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 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 an adapter for grid platforms to be connected. 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 an 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

* This paper is supported by Nation Science Foundation of China under grant 90412010

its testing results. Finally, we conclude and discuss future work.

2. Related Works

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).
- Combining FT-MPI with H2O [11]: This work presents a framework used to bind FT-MPI and H2O technology. It provides a fault-tolerant MPI environment and owns the advantages of H2O.
- WSDM [12]: WSDM (*Web Service for Distributed Management*) is a specification for interoperation between various platforms based on web services. The greatest advantage is allowing administrators defining the relationships between different terms in terms of a common standard.

Comparing with all these 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. We design an 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:** $V=\{v \mid v$ is an entity in ontology$\}$
- **Definition 4. Property Set:** $P=\{p \mid v.p\ , v\in V\}$
- **Definition 5. Relationship Rule:** R=some rules based on semantics = $\{r_1, r_2, \ldots, r_n\}$, 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).

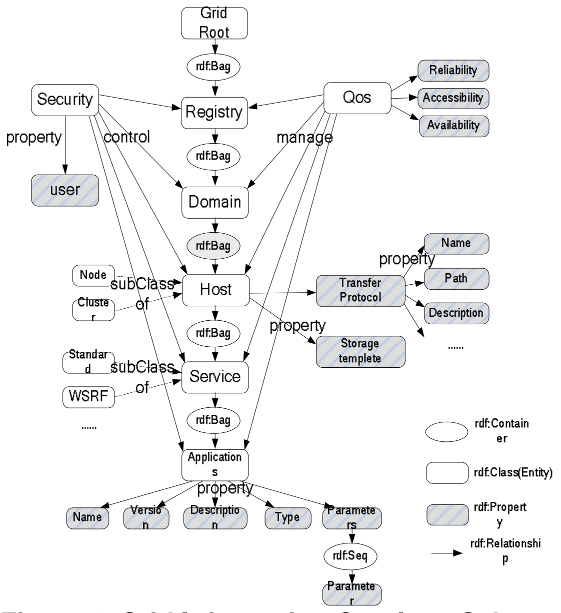
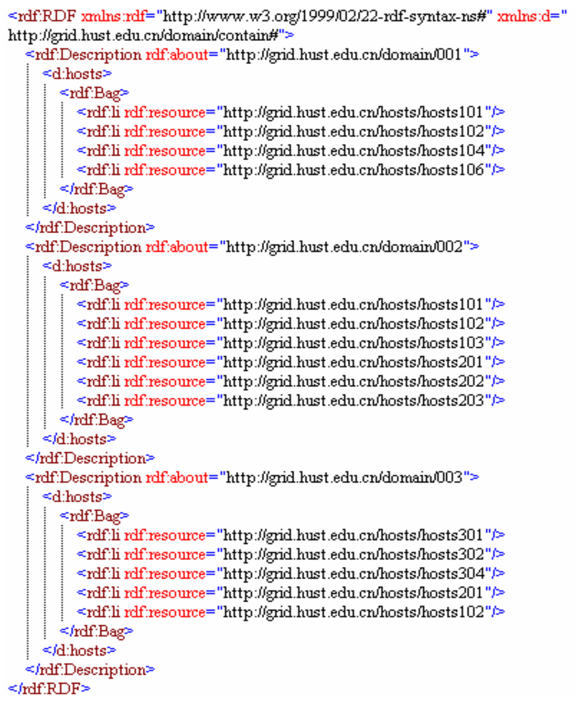


Figure 1. Grid Information Ontology Schema

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 hosts, 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 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.

$$\text{Application1.host=Host1} \rightarrow \text{(Application1, containedBy, Host1)} \tag{1}$$

$$\text{(Host1, containedBy, Service1), (Service1, containedBy, Domain1)} \rightarrow \text{(Host1, containedBy, Domain1)} \tag{2}$$

4. Interoperation Based on Ontology

The ontology is a uniform backbone for grid information interoperation. 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 intercommunication 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.

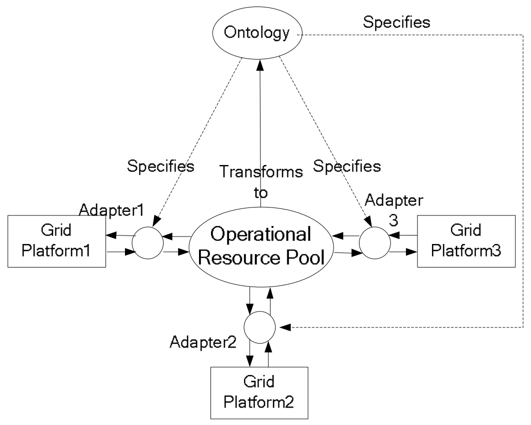


Figure 3. Common Access to Information Frame

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

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

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

exists(lack key entity): It returns true when lacking some key entities. (*key entity* means this entity can not be ignored 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 a grid platform to the standard operational resources.

```
Input: schema_mapping_list, Registry's recorded information
                                        (XML Streams/Files)
Output: ontology or failing report
begin
   get the schema_mapping_list and Registry's recorded information);
   call check-up-integrality function: check (schema_mapping_list));
    if (check(schema_mapping_list)=true) then
       begin
       construct ontology;
       construct the corresponding resources with schema_mapping_list;
       construct Registry's recorded information;
       end
    else    // check(schema_mapping_list)=false
       begin
       report the failure of check-up-integrality;
       end
end
```

Figure 4. Grid Platform to Operational Resources

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

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

```
Input: some ontology and mapping_list
Output: XML files according to the schema of some target grid platform's
Registry
begin
    get the schema_mapping_list;
    get Registry's recorded information;
    call check (schema_mapping_list)//check-up-integrality function
    if (check(schema_mapping_list)=true) then
        begin
            create XML files with ontology and schema_mapping_list);
        end
    else    // check(schema_mapping_list)=false
        begin
            report the failure of check-up-integrality);
        end
end
```

Figure 5. Operational Resources to Grid Platform

```
Input: schema_mapping_list
Output: true or false
begin
    foreach(terms in ontology schema) do
        begin
        get one term: t;
        if(t has not been mapped in schema_mapping_list) then
            begin
            if (inferable(term)=true) then // the term is inferable
                begin
                    infer(term);
                    add the mapping to schema_mapping_list);
                end
            else if (exists(lack key entity)=true) then
                begin
                    return false;
                end
            else
                begin
                set a default value to the term;
                end
            end
        else
            begin
                return false;
            end
        end
    return true;
end
```

Figure 6. Check-up-integrality function: *check(schema_mapping_list)*

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

1) The administrators of the two target grid platforms submit their own information schemas respectively according to some prescriptive formats.

2) The corresponding adaptors analyze the submitted schemas and check their integralities.

3) The adaptors fuse the semantic discrepancies between different grid platforms by transforming them to common operational resources and wait for request events.

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 to map semantics by a set of adapters. We give some sample 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.

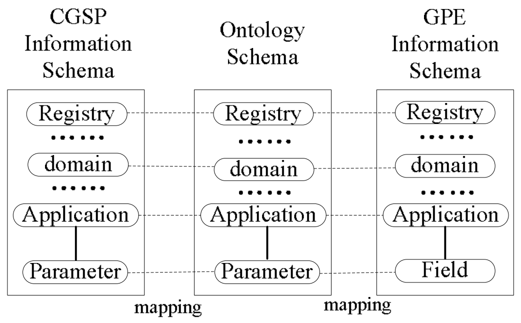
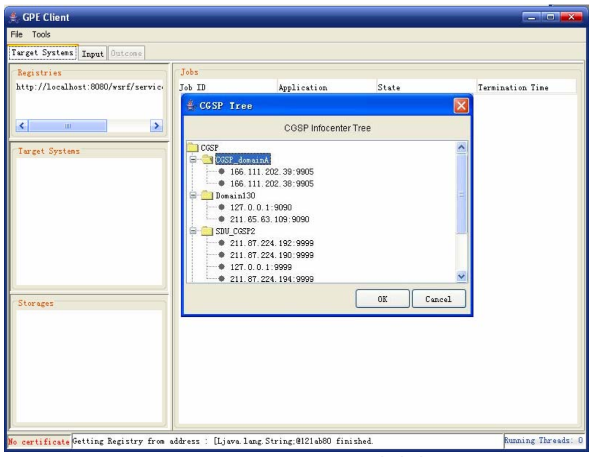


Figure 7. Mapping between Grid Platform Information Schemas and Ontology Schema

Figure 8 shows the visual panel of GPE's administrator in which CGSP's information is displayed.



```
http://localhost:8080/wsrf/servic

CGSP
  CGSP_domainA
    166.111.202.39:9905
    166.111.202.38:9905
  Domain130
    127.0.0.1:9090
    211.65.63.109:9090
  SDU_CGSP2
    211.87.224.192:9999
    211.87.224.190:9999
    127.0.0.1:9999
    211.87.224.194:9999

No certificate  Getting Registry from address : [Ljava.lang.String;@121ab80 finished.  Running Threads: 0
```

Figure 8. The Information of CGSP Displayed in the Visual Panel of GPE's Administrator

5.2 Performance Evaluation

The performance of our grid information interoperation platform mainly lies on the number of resources (including entities and properties) to be created in semantics translation. 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 1 shows the details.

Table 1. The Precondition of the Target Grid Platform Information

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 9.

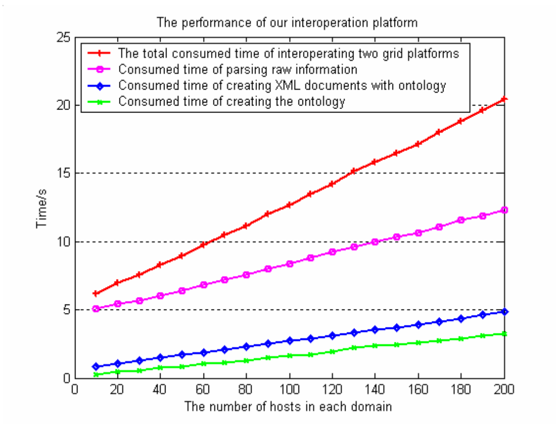


Figure 9. The Performance of Our Interoperation Platform

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

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

$$O(n) = K_1 \cdot n_3 + K_2$$
$$(K_1 = n_1 \cdot n_2 \cdot c_3 + n_1 \cdot n_2 \cdot n_{4\max} \cdot c_4 + n_1 \cdot n_2 \cdot n_{4\max} \cdot n_{5\max} \cdot c_5 \qquad (4)$$
$$K_2 = n_1 \cdot c_1 + n_1 \cdot n_2 \cdot c_2)$$

From Formula (4) and Figure 9, 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 an 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 will 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", *Proceedings of International Parallel and Distributed Processing Symposium (IPDPS'05)*, 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.source forge.net