ISSN 2047-3338

Ontology-Based Semantic Matchmaking of Grid Services Using Parameter Matching Algorithm

Saurabh Saxena[1], Rajeev Kumar[2] and Dr. M.K. Sharma[3]

[1]Department of Computer Science, Indraprastha Institute of Technology, Gajraula (J.P. Nagar-India)
[2] Department of Computer Science, College of Engineering Teerthanker Mahaveer University Moradabad, India
[3]Department of Computer Science, Amrapali Institute, Haldwani (Uttarakhand-India)

Abstract– **The fundamental problem the Grid research and development community is seeking to solve is how to coordinate distributed resources amongst a dynamic set of individuals and organizations in order to solve a common collaborative goal. The problem of service discovery in a Grid environment arises through the heterogeneity, distribution and sharing of the resources in different virtual organizations. This paper proposes a service discovery framework which is based on semantics. It gives an example of the Grid Job Submission Service written in DAML-S in order to show how service ontologies are implemented. This semantic approach allows a more flexible and dynamic matching mechanism based on semantic descriptions stored in ontologies.**

Index Terms– **Ontology, Matching Mechanism, Grid Service Discovery and Algorithm**

1. INTRODUCTION

THE term, grid computing, has become one of the latest buzzwords in the IT industry. Grid computing is an innovative approach that leverages existing IT infrastructure to optimize compute resources and manage data and computing workloads. According to Gartner, "a grid is a collection of resources owned by multiple organizations that is coordinated to allow them to solve a common problem." Gartner further defines three commonly recognized forms of grid: Computing grid - multiple computers to solve one application problem Data grid - multiple storage systems to host one very large data set Collaboration grid - multiple collaboration systems for collaborating on a common issue. Grid computing is not a new concept but one that has gained recent renewed interest and activity for a couple of main reasons: IT budgets have been cut, and grid computing offers a much less expensive alternative to purchasing new, larger server platforms. Computing problems in several industries involve processing large volumes of data and/or performing repetitive computations to the extent that the workload requirements exceed existing server platform capabilities.

Some of the industries that are interested in grid computing include: life sciences, computer manufacturing, industrial manufacturing, financial services, and government.

SAS views grid computing as a means to apply the resources from a collection of computers in a network and to harness all the compute power into a single project, for example. Grid computing can be a cost effective way to resolve IT issues in the areas of data, computing and collaboration; especially if they require enormous amounts of compute power, complex computer processing cycles or access to large data sources. SAS additionally believes that grid computing needs to be a secure, coordinated sharing of heterogeneous computing resources across a networked environment that allows users to get their answers faster.

II. RELATED WORK

Searching by traditional search engines based on keywords [1] has its own problems. As a result they do not check semantic of search objects and simply treat them as character strings. A lot of irrelevant information will be returned to the user as long the keywords appear somewhere in their files. The literature [2] proposes a meta search engine called “guided google”, that is built using the Google web services. This search engine guides and allows the user to view the search results with different perspectives, which achieved through simple manipulation and automation of Google functions.

However, the functionalities provided in this engine are based on “combinational keyword search” and it neither supports semantic description nor performs semantic search. The literature [3] addresses the problem of resource description in the context of resource broker to broker for resources described by several Grid middleware including Unicore. In this research work, we propose a semantic grid architecture using PEG that addresses the issue of semantic description through OWLS plug-in and discovery of services. The PEG allows the service provider for providing semantic description of grid services using OWLS editor that comes as a plug-in to protégé ontology editor. The Parameter Matchmaking Algorithm proposed in this paper compares the IOF of advertised and requested service and determines the degrees of match. This degree of match reveals how similar they are. Algernon inference engine [8] is used to retrieve IOF parameters from the advertised service ontology in PEG.

We also propose an semantic grid architecture in which protégé editor is integrated with Globus middleware making possible for semantic descriptions of Grid Services.

III. SEMANTIC GRID SERVICES USING PEG

The PEG addresses the demands of a single toolkit to build Grid infrastructure as well as for semantic description and representation of services. Currently, the concept of ontology is widely used for conceptual representation of a particular domain and Web Ontology Language (OWL) is used to develop ontology as a concept. Ontologies are used to capture knowledge of a domain of interest. Ontology describes the concepts in the domain and also the relationships that hold between those concepts [4].

The most recent development in standard ontology language is OWL from the World Wide Web Consortium. It makes it possible to describe concepts and it also provides a richer set of operators such as *and, or* and *negation*. It is based on different logical models which make it possible for concepts to be defined as well as described [5]. Further, the logical model allows the use of reasoner which can check whether all of the statements and definitions in the ontology are mutually consistent or not and it can also recognize which concept falls under which definition. Protégé editor is an integrated software tool used by system developers and domain experts to develop knowledge based systems [6].

This tool is widely used to create ontology in many applications. Protégé editor has Algernon inference engine has plug-in which facilitates direct interaction with Protégé knowledge bases (KBs) and it supports access to multiple concurrent KBs. Algernon commands not only retrieve and store slot values, but can also modify the ontology by executing Algernon queries in PEG, the Protégé editor along with OWLS plug-in is integrated with GT to address the demand of single toolkit for semantic description and representation of services by creating service ontology and its capability is extended to enable semantic description and representation of services by creating service ontology. In this paper, we propose a five layered architecture using PEG as middleware for semantic description and discovery of services.

VI. LAYERED ARCHITECTURE OF SEMANTIC GRID USING PEG

The five layered architecture proposed for semantic grid services is shown in Figure 1. Each layer shares the behavior of the underlying component layers and the same is explained below.

Fabric layer The Fabric layer deals with the resources available in grid environment and defines the interface to local resources, which may be shared. This includes computational resources, data storage, networks, catalogs, software modules, and other system resources. Grid Middleware Services This layer incorporates Grid Middleware and we use PEG as Grid Middleware in this research work. It also consists of required protocols for authentication and authorization which are implemented using Grid Security.

A. Knowledge Services Layer

Running on top of the high level grid service layers, the knowledge service layer can provide knowledge discovery from a huge amount of data. This layer is domain oriented and usually consists of service ontology built using protégé editor. The parameter matchmaking algorithm proposed in this paper is implemented in this layer that performs matchmaking of services based on IOF parameters.

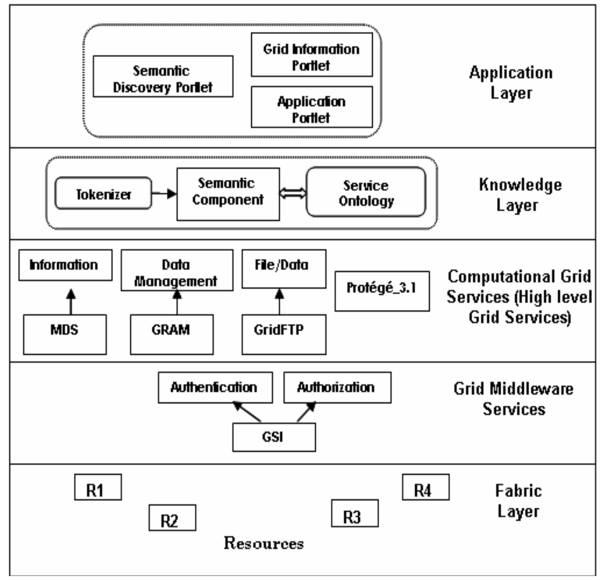

Fig. 1. A layered architecture for semantic grid services using PEG

B. Application Layer

The application layer enables the use of resources in a grid environment through various collaboration and resource access protocols. The semantic port-let present at this layer enables the service provider to register the service into the MDS registry and it prompts the provider to describe the service semantically using Protégé editor. The port-let also enables the service requester to submit the query and semantic retrieval of information from the service ontology using the proposed matchmaking algorithm. In addition to that, this layer may also consist of various application port-lets to use grid resources.

V. PARAMETER MATCHMAKING ALGORITHM

Matchmaking refers to capability matching which means to compare the requested service description with the advertised service descriptions [9]. In this paper, we use IOF to express the capability of a service. The goal of this capability comparison is to obtain information on how similar they are [9]. This degree of similarity is used to determine degrees of match between the advertised services descriptions takes all the inputs and the outputs into account [9].

In this research work, the proposed algorithm computes various matching degrees of service advertisement (A) and request (R) by successively applying different filters. The comparison is based on three parameters of the service namely the Inputs, Outputs and Functionalities (IOF). The service ontology that clearly describes IOF of the service is created using protégé editor of PEG to enable effective matchmaking of services. The algorithm semantically compares the IOF of the requested services with that of the advertised ones and computes various degrees of matches as listed below:

Exact Match: Here the advertised IOF of the service are exactly matches with that of requested service. We use Rank 1 to this match. In our context,

$A(IOF) \equiv R(IOF) \rightarrow \{ A(I) \equiv R(I) \cap A(O) \equiv R(O)$
$\cap A(F) \equiv R(F) \}$

Plug-in Match: This match occurs if A describes greater capability than that R requires. We use Rank 0.75 to this match. In our context,
A(IOF) ≥ R(IOF) → { A(I) ≥ R(I) U A(O) ≥ R(O) U A(F) ≥ R(F) }

Subsume: This match occurs if R requests greater capability than that R requires.
A(IOF) ≤ R(IOF) → { A(I) ≤ R(I) U A(O) ≤ R(O) U A(F) ≤ R(F) }

Intersection: This filter reveals that not all the capabilities requested by the service matching with the advertised capabilities. We use Rank 0.25 to this match.

Disjoint: The requested service R does not match with the described service A according to any of the above filters. In our context,
A(IOF) ≠ R(IOF) → { A(I) ≠ R(I) U A(O) ≠ R(O) U A(F) ≠ R(F) }

Rank 0 is used for this match. The proposed Parameter Algorithm implemented in this paper is given below:

Algorithm Parameter Matchmaking Algorithm

```
Input: Advertised_ontology A, Request R
Output: Degree_of_Match M
Rank: input_rank, output_rank, functionality_rank
Parse A into A(I1,I2…Im),A(O1,O2,..On) and A(F1,F2…Fp)
Parse R into R(I1,I2…Ir),R(O1,O2,..Os) and R(F1,F2…Ft).
c1=0, c2=0, c3=0,i=0, j=0
for each parsed A(I1,I2…Im),A(O1,O2,..On), A(F1,F2…Fp)
do
if A(Ii) ≡ R(Ij) then c1++;
if A(Oi) ≡ R(Oj) then c2++;
if A(Fi) ≡ R(Fj) then c3++;
end if
end for
input_rank=compute_intermediaterank(m,c1,r)
output_rank=compute_intermediaterank(n,c2,s)
functionality_rank=compute_intermediaterank(p,c3,t)
M=leastof(input_rank,output_rank,functionality_rank)
Rank compute_intermediate(i,c,j)
{
if(i==c==j) then R=1;
if(i>c=j), then R=0.75;
if(i=c<j), then R=0.50;
if(i>c<j), then R=0.25;
if(i!=c!=j), then R=0;
}
```

IV. IMPLEMENTATION

The parameter matchmaking algorithm is implemented in knowledge layer of the proposed architecture using java language in this paper. The java implemented Algernon packages are used to query the ontology knowledge base. The package offers several java APIs with which various queries can be executed. The java implemented tokenizer extracts IOF from the service requester's query by eliminating unwanted information from the query which is then compared with that of the advertised service and computes the degrees of matches. The algorithm starts with extracting IOF from the advertised service by executing appropriate algernon queries over service ontology described in PEG. The tokenizer implemented in the semantic component receives the service requester's query and identifies IOF. The algorithm will then go through four stages as shown in Figure 2 to compute the degrees of match. The matchmaking module then performs semantic comparison of IOF of the requested service R(IOF) with that of advertised A(IOF) service individually in three stages and computes three intermediate ranks namely Ir, Or, and Fr as shown in the Fig 2. All the intermediate ranks are combined together in aggregate module and least rank is considered as the final rank.

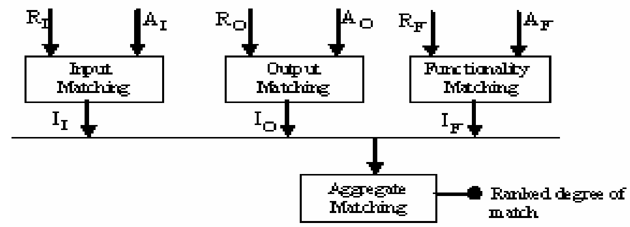

Fig. 2. Ranked degree of match

This final rank reveals the degrees of match and the requester is allowed to access service if the ranked degrees of match are neither intersection nor disjoint.

A. Semantic Grid Portal

A Grid portal that consists of several port-lets to provide required user interface for semantic description and discovery of services is developed. It provides necessary interface for the service providers to register their grid service and to describe it semantically. It also provides interface for the service requesters to submit their queries and to perform matchmaking of services.

The Service Oriented Architecture model of the proposed architecture for semantic grid service is shown in Fig. 4, 5.

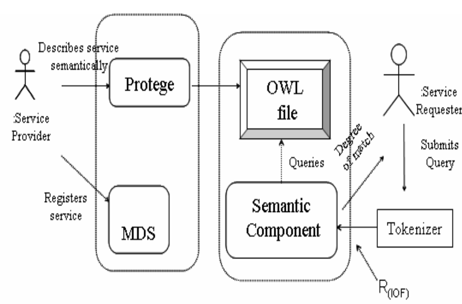

Fig. 4. Service oriented architecture of semantic grid using PEG

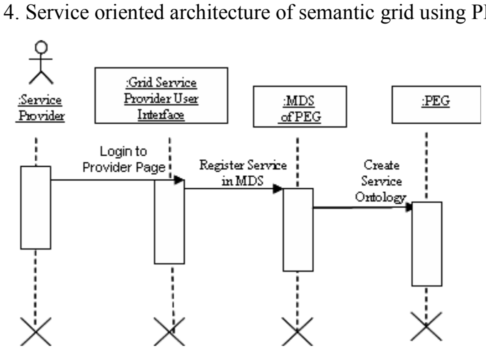

Fig. 5. Sequence diagram of service provider

IIV. CONCLUSION

In this research paper, we extended the capability of Globus Toolkit 4.0 by integrating Protégé ontology editor in it. This feature facilitates the Grid Service Providers to describe their services semantically through OWLS editor. The semantic description of services enables semantic discovery of services. A matchmaking algorithm is proposed that performs semantic matchmaking of services on the basis of IOF parameters. The user interface for semantic description and retrieval is developed as a portal enabling the user to interact easily with the grid environment. Several Grid service have been implemented and described semantically using PEG. The proposed Architecture using Parameter Matchmaking Algorithm can be applied for any specific applications enabling the users to access grid comfortably.

REFERENCES

[1]. Foster, I. and Kesselman, C. (eds), “The Grid: Blueprint for a New Computing Infrastructure”, Morgan Kaufmann, 1999, 259-278.

[2]. Berners-Lee,T., Hendler,J. and Lassila, O. “The Semantic Web”, Scientific American, May 2001.

[3]. David De Roure, Nicholas R. Jennings and Nigel R. Shadbolt, “The Semantic Grid: Afuture e-Science Infrastructure”, Grid Computing – Making the Global Infrastructure a reality, John Wiley & Sons, Ltd, 2003.

[4]. Lei Li,Ian Horrocks, “A Software Framework For Matchmaking Based on Semantic Web Technology”, WWW2003, May 20-24,2003.

[5]. Stumme, G., Ehrig, M., Handschuh, S., Hotho, A., Maedche, A., Motik, B., Oberle, D., Schmitz, C.,Staab, S., Stojanovic, L., Stojanovic, N., Studer, R., Sure, Y., Volz, R., Zacharias, V., “The Karlsruhe view on Ontologies”, Technical report, University of Karlsruhe, Institute AIFB (2003).

[6]. http://algernon-j.sourceforge.net/tutorial/

[7]. Grigoris Antoniou and Frank van Harmelen, “A Semantic Web Primer", The MIT Press, 2004.

[8]. YeZhang and William Song, “Semantic Description and Matching of Grid Services Capabilities”.

[9]. Micheal C. Jaeger, Gregor Rojec-Goldmann, Christoph Liebetruth and Kurt Geihs, “Ranked Matching for Service Descriptions using OW”.