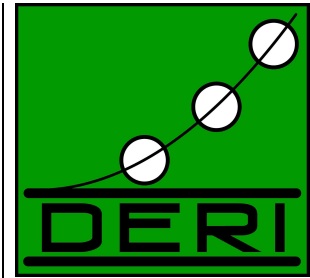


DERI – DIGITAL ENTERPRISE RESEARCH INSTITUTE



## A SEMANTIC MATCHMAKER SERVICE ON THE GRID

Andreas Harth      Yu He  
Hongsuda Tangmunarunkit      Stefan Decker  
Carl Kesselman

DERI TECHNICAL REPORT 2004-05-18

MAY 2004

DERI – DIGITAL ENTERPRISE RESEARCH INSTITUTE

**DERI Galway**  
University Road  
Galway  
IRELAND  
[www.deri.ie](http://www.deri.ie)

**DERI Innsbruck**  
Technikerstrasse 13  
A-6020 Innsbruck  
AUSTRIA  
[www.deri.ie](http://www.deri.ie)



DERI TECHNICAL REPORT  
DERI TECHNICAL REPORT 2004-05-18, MAY 2004

A SEMANTIC MATCHMAKER SERVICE ON THE GRID

Yu He<sup>1</sup>, Hongsuda Tangmunarunkit<sup>1</sup>, Carl Kesselmann  
Andreas Harth<sup>1 2</sup>, Stefan Decker<sup>1 2</sup>

**Abstract.** A fundamental task on the Grid is to decide what jobs to run on what computing resources based on job or application requirements. Our previous work on ontology-based matchmaking discusses a resource matchmaking mechanism using the Semantic Web technologies. We extend our previous work to provide dynamic access to such matchmaking capability by building a persistent online matchmaking service. Our implementation uses the Globus Toolkit for the Grid service development. It also exploits the monitoring and discovery service in the Grid infrastructure to dynamically discover and update resource information. A schema translator is developed to translate the various formats used in a heterogeneous Grid environment into our resource ontology. We describe the architecture of our semantic matchmaker service.

**Keywords:** Grid service, Semantic Web, networking and distributed web applications, resource allocation, resource selection.

---

<sup>1</sup>USC Information Sciences Institute, 4676 Admiralty Way, Marina del Rey, CA 90292  
E-mail: {yuhe, hongveda, carl}@isi.edu

<sup>2</sup>Digital Enterprise Research Institute Galway, National University of Ireland Galway  
University Road Galway, Ireland E-mail: {andreas.harth, stefan.decker}@deri.ie

## **Contents**

<b>1</b>	<b>Introduction to the Matchmaking Problem</b>	<b>1</b>
<b>2</b>	<b>Ontology-based Matchmaker (OMM)</b>	<b>1</b>
<b>3</b>	<b>OMM Service Architecture</b>	<b>2</b>
3.1	Matchmaking Module . . . . .	2
3.2	Request Handling . . . . .	2
3.3	Resource Discovery . . . . .	3
<b>4</b>	<b>Conclusion</b>	<b>3</b>

## 1 Introduction to the Matchmaking Problem

Grid is an emerging technology for enabling resource sharing and coordinated problem solving in dynamic multi-institutional virtual organizations. In the Grid environment, shared resources and users typically span different organizations. The resource matching problem in this environment involves assigning resources to tasks in order to satisfy task requirements and resource policies. These requirements and policies are often expressed in disjoint application and resource models, forcing a resource selector to perform semantic matching between the two.

Grids are used to join various geographically distributed computational and data resources, and deliver these resources to heterogeneous user communities. These resources may belong to different institutions, have different usage policies and pose different requirements on acceptable requests. Grid applications, at the same time, may have different constraints that can only be satisfied by certain types of resources with specific capabilities. Before resources can be allocated to run an application, a user or agent must select resources appropriate to the requirements of the application. We call this process of selecting resources based on application requirements "resource matching". In a dynamic Grid environment, where resources may come and go, it is desirable and sometimes necessary to automate the resource matching to robustly meet application requirements.

## 2 Ontology-based Matchmaker (OMM)

Existing resource description and resource selection in the Grid is highly constrained. Traditional resource matching, as exemplified by the Condor Matchmaker, is done based on symmetric, attribute-based matching. In these systems, the values of attributes advertised by resources are compared with those required by jobs. For the comparison to be meaningful and effective, the resource providers and consumers have to agree upon attribute names and values. The exact matching and coordination between providers and consumers make such systems inflexible and difficult to extend to new characteristics or concepts. Moreover, in a heterogeneous multi-institutional environment such as the Grid, it is difficult to enforce the syntax and semantics of resource descriptions.

In our ontology-based matchmaker, we employ a flexible and extensible approach for performing Grid resource selection, that, unlike the traditional Grid resource selectors, uses separate ontologies to declaratively describe resources and job requests using an expressive ontology language. Instead of exact syntax matching, our ontology-based matchmaker performs semantic matching using terms defined in those ontologies. The loose coupling between resource and request descriptions remove the tight coordination requirement between resource providers and consumers. In addition, our matchmaker can be easily extended, by adding vocabularies and inference rules, to include new concepts about resources and applications and adapted the resource selection to changing policies.

In our previous work, we have designed and prototyped our matchmaker using TRIPLE to use ontologies encoded in W3C's Resource Description Format (RDF) and rules (based on Horn logic and F-Logic) for resource matching. Resource descriptions, request descriptions, and usage policies are all independently modeled and syntactically and semantically described using RDF schema. Finally, we utilize inference rules for reasoning about the characteristics of a request, available resources, and usage policies to appropriately find a resource that satisfies the request requirements.

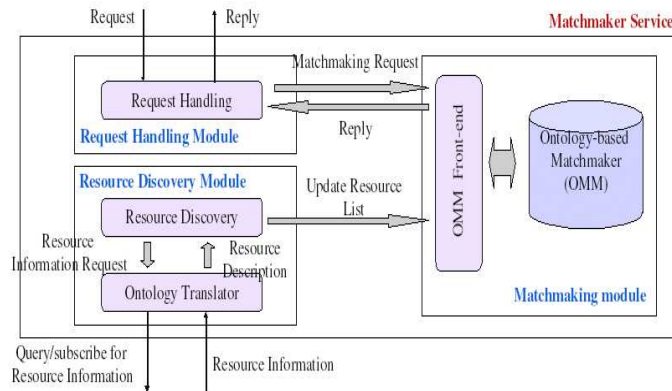


Figure 1: Architecture of the Ontology-based Resource Matchmaker Service (OMMS).

### 3 OMM Service Architecture

Apart from providing the core matchmaking functionality, the matchmaker has additional requirements: the service must be available in continuous operation, and other Grid components must be able to access that functionality online. Our previous prototype finds matches between resources and requests based on static information, *e.g.*, by reading resource and request descriptions from RDF files, performing the matching, and then exiting. The descriptions about all available resources must be reloaded every time a request is posed. Moreover, there is no mechanism to dynamically discover or update the status of resources.

In this poster, we extend our previous work by developing means to access the ontology-based matchmaker using the Grid service technology. A Grid service is an extended Web service that conforms to the Open Grid Service Infrastructure (OGSI) specification. In our work, we utilize the Globus Toolkit to develop a Grid service. In addition to the semantic resource matching, the ontology-based matchmaker service (OMMS) supports dynamic resource discovery, and schema translation by mapping different resource advertisement schemas and encodings to our resource ontology. Our OMMS architecture comprises three modules: matchmaking, request handling, and resource discovery (Figure 1).

#### 3.1 Matchmaking Module

For the matchmaking component used in the Grid service, we build upon our previous work based on the TRIPLE query and inference system. TRIPLE is written in Java, and integrated with the C-based XSB system via the Java Native Interface (JNI). Our previous prototype is extended to return results as Java objects, which allows us to further process the results and return a list with matching computing resources to the request handling module. While running the ontology-based matchmaker as a Grid service, changes in the Grid environment must be detected and incorporated in OMM's knowledge base without interrupting the service. Therefore, TRIPLE is extended to allow for dynamic adding/retracting of facts and rules so that a stand-alone persistent service can be realized.

#### 3.2 Request Handling

A request is expressed using our request ontology. We design a persistent Grid service that can support multiple clients simultaneously. Since there is only one single TRIPLE instance that handles all requests,

we implement a synchronization mechanism to ensure an atomic operation for each request. The granularity of these operations is chosen so that the waiting time for each operation is short.

Clients submit requests in RDF, and receive a list of matching resources either in an outline form (*e.g.*, resource names) as strings or in a detailed form (*e.g.*, resource names and their capabilities) as RDF. Users can express their preference for matched resources by using a ranking function. The ranking function is an arithmetic expression expressed in terms of resource properties; for example,  $10 * CPU\text{ClockSpeed} + PhysicalMemory$ . Clients can indicate the number of returned resources which will be sorted based on their ranking values.

### 3.3 Resource Discovery

As a key part of OMMS, the resource discovery component dynamically collects resource information from multiple sources, transforms the information into our resource ontology, and updates the backend knowledge base accordingly. Since different Grid resource providers might express their capabilities using different schemas and encoding mechanisms (*e.g.*, XML, LDAP), an ontology translator is developed to translate the various formats used in a heterogeneous Grid environment into our resource ontology. Our design goals of this module include collecting various categories of resources, updating with least communication overhead without service interruption.

The resource discovery module exploits the monitoring and discovery services equipped in the Grid infrastructure to dynamically discover and update resource information. For those resource information providers that support subscription/notification services, such as Grid Information Service (GIS), OMMS subscribes to those providers to obtain asynchronous updates. With this mechanism, the OMMS will be notified only when there are changes in the status of the resources, and hence reduce the communication overhead between OMMS and providers. For those information providers that do not support subscription service, we deploy a polling mechanism by periodically querying the providers for up-to-date resource information. The polling period is chosen to obtain a balance between communication overhead and data consistency. To reduce communication overhead, upon obtaining latest information about the status of Grid resources, the resource discovery component submits only incremental changes to the OMM backend (*i.e.*, the matchmaking module).

Since OMMS is persistent, resource information kept by OMMS could be active for the entire lifetime of the service. To efficiently adapt to the changing Grid computing environment, the resource information is maintained using a softstate mechanism by associating a lifetime (*e.g.*, the validity period of the information) with each resource description. If the description is not updated before the lifetime expires, the resource is retracted from the knowledge base. Finally, OMMS enforces atomic operation during each resource update to avoid conflicts in the shared knowledge base.

## 4 Conclusion

In this poster, we have presented our work that exploits two novel technologies—the Grid and Semantic Web technologies—to provide a persistent online service for solving resource matching problem in the Grid. The existing system is designed to be application or domain independent. However, higher-level application models can be built on top of our existing system. As an ongoing research, we are expanding the request ontology to include application-level description as well as their performance models, allowing users to describe their requirements in their own domain-specific terms. Depending on the request, additional information might be needed. For example, the sizes of input files are needed to infer the disk space

requirement associated with the computing resource. In addition to the ontology expansion, we are extending the matchmaker service to dynamically gather additional information from other knowledge sources (such as the Metadata Catalog Service) and incorporate that into the matchmaking procedure.