An Evaluation of Discovery approaches in Grid and Web services Environments *

Ioan Toma Kashif Iqbal Matthew Moran Dumitru Roman
Thomas Strang
Dieter Fensel

Digital Enterprise Research Institute (DERI), Galway, Ireland and Innsbruck, Austria.
`<firstname>.<lastname>@deri.org`

Abstract: The challenge of finding services or resources in Web services and Grid environments has recently been the subject of a lot of attention. Different solutions to this problem were proposed, each with its specific model and realization. Although all solutions address the same problem, it is very difficult for a non-expert and even for an expert in the field to decide if one solution is better then another and why. In this paper we propose a systematic set of criteria, a framework, that can help in the evaluation of different discovery approaches. We exemplify the use of our framework on some of the most relevant discovery approaches in Web services and Grid areas.

1 Introduction

Web services and Grid Computing are technologies with a great impact in industry and academic. Grid computing aims to provide the computational power and data management infrastructure necessary to support the collaboration of people, together with data, tools and computational resources [FK99]. The most common problems that Grid addresses are computationally hard and data intensive problems in science and engineering. Grids offer a solution to these problems by joining geographically distributed computational and data resources. These resources are delivered to heterogeneous user communities. The Grid provides the protocols, services and software development kits needed to enable flexible, controlled resource sharing on a large scale. Sharing in Grid is, necessarily, highly controlled with resource providers and consumers defining clearly and carefully just what is shared, who is allowed to share, and the conditions under which sharing occurs.

Web services, on the other hand, try to solve the problem of enterprise application integration and automation of business processes by making use of the Web as a global infrastructure for distributed computation [ACKM03]. They offer a new level of automation in eWork and eCommerce, where fully open and flexible cooperation can be achieved, on-the-fly, with low programming costs [FB02]. To realize this, a suite of protocols: UDDI[1],

*The work is funded by the European Commission under the project *Adaptive Services Grid.* (ASG)
[1] http://www.uddi.org/

WSDL[2], and SOAP[3] are in common use. In this way software applications can be accessed and executed via the Web based on the idea of Web services.

One important problem common to both domains mentioned above is how to find an item (resource, service) that can fulfill the requested functionality. An automated solution to this problem known as *discovery* would be a big step forward in Web services and Grid domains.

Recent efforts in Grid computing [CFF$^+$04] treat resources as Web services and for this reason discovery of resources becomes discovery of Web services in the latest Grid systems. Resources are basically wrapped by a Web service. Having this model in mind we evaluate discovery approaches for Web services and Grid environments, as they can be applied in each domain with some extensions or modifications.

The goal of this paper is to provide a set of criteria that can be used to evaluate the current effort in discovery of resources and services. The paper is structured as follows: Section 2 provides the terminology used in this paper. Section 3 defines a framework used in the evaluation of discovery approaches. Section 4 surveys some of the most relevant discovery approaches from both Web services and Grid areas and evaluates them based on the evaluation framework previously defined. Section 5 summarizes the evaluated approaches and abstracts the common best-practices from the various approaches. Finally, Section 6 presents the related work and concludes the paper.

2 Terminology

In order to evaluate a process or a mechanism, one has to have a clear understanding about the settings, the entities manipulated and the functionality of the process or mechanism. Some questions automatically pop-up when trying to define such an evaluation mechanism for service and resource discovery in Grid and Web services environments.

What is Grid? There are many interpretations of the term Grid [Grab], [Graa]. We don't provide a new definition of the term, but instead we use one of the most common adopted and used definition provided in Open Grid Services Architecture [FKNT02]. According to [FKNT02], the *Grid provides the protocols, services and software development kits needed to enable flexible, controlled resource sharing on a large scale within a Virtual Organization.* A Virtual Organization is a dynamic collection of individuals, institutions and resources bundled together in order to share resources as they tackle common goals.

What is a Service? The term *service* has been lately semantically overloaded [Pre04]. Communities like: business science, information science and computer science have a different understanding of what a service is [BGOA04]. We adopt the definition provided in the conceptual model architecture for semantic Web services [Pre04]. According to [Pre04] a *service is a provision of value in some domain* (not necessarily monetary value).

[2]http://www.w3.org/TR/wsdl
[3]http://www.w3.org/TR/soap/

What is a Web service? Web services have emerged as a promising technology that offers a standard way to access and integrate functionalities. They are loosely coupled software components published, located and invoked across the Web. For the Web service concept definition we again adopt the description provided in [Pre04]. A *Web service* is defined as a *computational entity accessible over the Internet (using Web service Standards and Protocols)*

What is a Resource? Resources are important concepts in a Grid environment. They form the *low level entities that are accessed and used to fulfill a user request*. Different resources can have the same functional capabilities but they may have different access policies associated, different time access, etc. As was pointed out before, in the latest approaches in Grid [CFF$^+$04] resources are treated as Web services and for this reason discovery of resources becomes discovery of Web services in the latest Grid systems.

What is Discovery? Discovery is an important process in both Grid and Web services environments. Very briefly it can be described as *the process that takes as input a user request and returns a list of resources or services that can possibly fulfil the given request*.

3 Evaluation Framework

With the terminology defined previously in mind we propose an evaluation framework for discovery approaches in Grid and Web services environments. This framework is then used in next sections to evaluate some of the most prominent approaches from Grid and Web services areas.

Our framework contains a set of criteria that we believe are the most relevant when performing evaluation of discovery approaches. We enlist and explain each of the criteria below:

- **Query Language and Advertising Language**
  Language support for query formulation as well as service and resource description is crucial for the discovery process. It is important to evaluate how expressive is the language and how easy it is to formulate queries and advertisements which are used during the discovery process. Special attention should be given to the semantic support that the language provides to express different aspects of advertisements and queries.
- **Scalability**
  Scalability is an important issue when talking about systems that have to accommodate changes in users, resources, etc. It is important to analyze how a system reacts when there is a grow in one or more of its dimensions. Another aspect that must be considered is how the scalability of the subsystems, or related systems, affects the scalability of the whole system. The scalability of a discovery mechanism, is for example heavily influenced by the scalability of the sub-components such as: query processor, storage, etc.
- **Reasoning support**

The provision of an automatic discovery mechanism can be significantly improved if machine processable descriptions are provided. These descriptions are further checked if they matched against each other(request versus services). New knowledge can be inferred based on existing facts (domain ontologies and domain background knowledge) and this knowledge can be further used during the discovery process. Therefore it is important to identify the reasoning support provided to enable the discovery process. Of course, this criteria is very closely related to *Query Language and Advertising Language* mentioned above.

- **Matchmaking vs. Brokering**
  When talking about the interactions between parties during the process of discovery two approaches can be distinguished (See [73]):

  1. *Matchmaking:*
     The entity which performs the matching determines if the request and the advertisements match but does not interfere in the next step which is the interaction between matched request and advertisements.
  2. *Brokering:*
     The entity which performs the matching determines if the request and the advertisements match and furthermore controls the interaction between matched request and advertisements.

  Many of Grid approaches for discovery distinguish between these two facets of the discovery process. Therefore it is important to evaluate an approach based on the support that it provides for pure matchmaking or brokering or both.

- **Mediation Support**
  Mediation provides an intermediary services, linking both data resources and application programs. Due to the heterogeneity of the environments (different data models, different protocols, etc.) where discovery has to be performed a mediation support is required. Therefor we add mediation support as another criteria to the evaluation framework.

4 Survey and Evaluation

We use the framework defined above to evaluate some of the most prominent approaches in Grid and Web services environments with respect to discovery. Due to space limitations we consider for evaluation only the discovery mechanisms provided by the following initiatives: *Unicore* [Rom02], *Globus* [FK97], *Ontology Based Matchmaker* [TDK01], *Web Service Modeling Ontology* [RLK04], [KLe04], *OWL-S approaches* [PKPS02].

UNICORE (**U**niform **I**nterface to **CO**mputer **RE**sources) [Rom02] is a Java based environment for secure and seamless access to remote supercomputers. The design goals for UNICORE include an open architecture based on the concept of an abstract job, consistent security architecture, minimal interference with local administrative procedures, exploitation of existing and emerging technologies, a zero administration user interface through a standard Web browser and Java applets. The UNICORE client enables the user to create, submit, and control jobs from any workstation or personal computer using the Internet. Third-party components, such as Globus [FK97], can be integrated into the UNICORE framework to extend its functionality. Resource discovery in UNICORE Grid architecture is accomplished by a GRIP Resource Broker component [BFM04] capable of locating resources on both UNICORE and Globus Toolkit based Grids, developed. The broker take workflows described in the UNICORE AJO framework and brokered the sites on EUROGRID[4]. UNICORE works with a conceptual representation of the job called Abstract Job Object, shortly AJO, which can be carried out on various computational and data services at collaborating sites.

- **Query Language and Advertising Language**
  The AJO is the basis for the platform and site neutral specification of requests for computational, data, and software resources. The object-oriented structure and syntax of the AJO make specification largely independent of hardware architecture, system software interfaces, and site-specific operational rules.
- **Scalability**
  There is a centralized broker which can be decentralized with the support of other brokers. The brokers follow hierarchical organization.
- **Reasoning support**
  No reasoning support for discovery is considered in UNICORE.
- **Matchmaking vs. Brokering**
  The EUROGRID/GRIP Resource Broker takes the workflows described in the UNICORE AJO framework and brokered the sites on EUROGRID to received offers from those sites which could enact the workflow and provide mechanisms for these sites to return tickets describing the quality of service policy they would offer.
- **Mediation Support**
  No mediation support for discovery is considered in UNICORE.

GLOBUS [FK97] provides software infrastructure for resource management of autonomous distributed systems with provisions for policy extensibility and co-allocation. A central element of the Globus system is the Globus Toolkit, which defines the basic services and capabilities required to construct Computational, Data or Service Grids. The basic services implemented by key components include: Grid Security Infrastructure (GSI), third party resource brokers, Grid Resource Allocation and Management (GRAM), data management (GridFTP, GASS), resource reservation (GARA), and communications. Globus is constructed as a layered architecture in which higher level services can be developed using the lower level core services [CFFK01]. Globus offers Grid information

[4] http://www.eurogrid.org

services via an LDAP-based network directory called Metacomputing Directory Services (MDS)[5].

Initial versions of MDS (version lower than 2.4) are based on LDAP network directory. Later versions of MDS (version higher than 3) and later releases are OGSA based. MDS comes with a default schema and information providers can populate the schema. However the default schema provided by MDS is not used in practice because is to simplistic (no notion of clusters for providers).Efforts like GLUE[6] try to provide a standard schema.

The MDS3 component of the Globus Toolkit Version 3.2 provides information about Grid resources for use in resource discovery, selection, and optimization. The MDS3 component is therefore a broad framework that includes any part of GT3 that generates, registers, indexes, aggregates, subscribes, monitors, queries, or displays Service Data in some way. The Index Service combines ServiceDataProviderExecution components with DataAggregation and ServiceGroup components to create a dynamic data-generating and indexing node, similar in concept to a MDS2 hierarchical GIIS. Index Services can be combined in a variety of topologies, useful in building Virtual Organizations. MDS follows both a push and pull protocols for resource dissemination. Higher-level tools such as resource brokers can perform resource discovery by querying MDS using LDAP and XML query protocols. Customers describe required resources through a resource specification language (RSL) that is based on a pre defined schema of the resources database. The task of mapping specifications to actual resources is performed by a resource co-allocator, which is responsible for coordinating the allocation and management of resources at multiple sites. RSL allows customers to provide very sophisticated requirements, but there is no possibility for resource providers to specify their constraints on customers. The MDS namespace is organized hierarchically in the form of a tree structure. Globus offers QoS in the form of resource reservation. It allows application level schedulers such as the Nimrod/G [BAG00] resource broker to extend scheduling capabilities. The resource brokers can use heuristics for state estimation while performing scheduling or re-scheduling whenever the status of the Grid changes. Globus is a Grid toolkit and thus does not supply scheduling policies; instead it allows third party resource brokers.

- **Query Language and Advertising Language**
  The Globus Resource Specification Language (RSL) provides a common interchange language to describe resources. The RSL provides the skeletal syntax used to compose complicated resource descriptions, and the various resource management components introduce specific *(attribute, value)* pairings into this common structure. Each attribute in a resource description serves as a parameter to control the behavior of one or more components in the resource management system.
- **Scalability**
  Globus supports distributed query based service discovery. The schedulers are in hierarchical organization. MDS can be decentralized using federation approach.
- **Reasoning support**
  No reasoning support for discovery is provided in Globus.

[5] http://www-unix.globus.org/toolkit/mds/
[6] www.hicb.org/glue/glue-schema/schema.html

- **Matchmaking vs. Brokering**
  Globus does not provide matchmaking or brokering services itself, rather metaschedulers can be used for brokering and matchmaking purposes like Condor-G[7], CSF, and Nimrod/G. Globus provides a monitoring and discovery service to provide both static and dynamic status information about resource properties.

- **Mediation Support**
  No mediation support for discovery is provided in Globus.

ONTOLOGY BASED MATCHMAKER (**OMM**) [TDK01] propose a flexible and extensible approach for performing Grid resource selection using an ontology-based matchmaking technique and algorithm. Unlike the traditional Grid resource selectors (like Condor-G) that describe resource and request properties based on symmetric flat attributes, separate ontologies (i.e., semantic descriptions of domain models) are created to declaratively describe resources and job requests using an expressive ontology language. Instead of exact syntax matching, ontology-based matchmaker performs semantic matching using terms defined in those ontologies. The loose coupling between resource and request descriptions removes the tight coordination requirement between resource providers and consumers. Matchmaker can be easily extended, by adding vocabularies and inference rules, to include new concepts (e.g., UNIX compatibility) about resources and applications and adapted the resource selection to changing policies. These ontologies can also be distributed and shared with other tools and applications. Matching between request specifications to resource capabilities is done in terms of rules. Similar to other matching systems, matchmaker provides the ability to describe properties and matching preference. Matchmaker also supports bi-lateral matching and gang-matching. The ontology-based matchmaker consists of three main components: *(1) Ontologies* that capture the domain model and vocabulary for expressing resource advertisements and job requests, *(2) Domain background knowledge*, capturing additional knowledge about the domain and *(3) Matchmaking rules*, defining when a resource matches a job description.

The background knowledge uses the vocabulary from the ontologies to capture background information. Matchmaking rules use both ontologies and background knowledge to match a request to resources. Ontology-based matchmaker is built on top of TRIPLE/XSB deductive database system, which is a centralized information store with no persistence. Resource discovery is through centralized queries composed using the vocabulary in the request ontology. The system aims to use Globus MDS as a persistent information store.

- **Query Language and Advertising Language**
  Resource descriptions, request descriptions, and usage policies are all independently modelled and syntactically and semantically described using a semantic mark-up language: RDF [Bec03], RDF Schema [BG04].

- **Scalability**
  Resource discovery is through centralized queries to a deductive database composed

[7] http://www.cs.wisc.edu/condor/condorg/

using the vocabulary in the request ontology.

- **Reasoning support**
  Domain background knowledge captured in terms of rules is added for conducting further deduction. Likewise, matchmaking procedures written in terms of inference rules are used to reason about the characteristics of a request, available resources and usage policies to appropriately find a resource that satisfies the request requirements. Additional rules can also be added to automatically infer resource requirements from the characteristics of domain-specific applications without explicit statements from the user.

- **Matchmaking vs. Brokering**
  OMM is a prototype system that currently provides only matchmaking services. In the next implementations the matchmaker will provide both matchmaking and brokering services. The brokering service will be built on top of matchmaking service.

- **Mediation Support**
  No mediation support for discovery is provided in OMM.

WEB SERVICE MODELING ONTOLOGY (**WSMO**) [RLK04] aims to develop an overall framework for Semantic Web services in order to support automated Web service discovery, selection, composition, mediation, execution, monitoring, etc. Every WSMO component description may include an extensible set of non-functional properties, based on the Dublin Core Metadata Set [WKLW98]. Two major design principles, inherited from WSMF [FB02] are applied in WSMO: *(1) Principle of maximal de-coupling*: all WSMO components are specified autonomously, independent of connection or interoperability with other components and *(2) Principle of strong mediation*: the connection and interplay between different components is realized by Mediators that resolve possible occurring heterogeneities between the connected components. WSMO defines four top-level notions related to Semantic Web services:

1. *Ontologies*: are the key to link conceptual real world semantics defined and agreed upon by communities of users. Ontologies define a common agreed upon terminology by providing concepts and relationships among the set of concepts.

2. *Goals*: are descriptions of user's desires; they represent the information space and state of the world after the execution of the service that would potentially satisfy the users desires.

3. *Web services*: are service descriptions for services that are requested by service requesters, provided by service providers, and agreed between service providers and requesters.

4. *Mediators*: address the handling of heterogeneities occurring between elements that shall interoperate by resolving mismatches between different used terminologies (data level), on communicative behavior between services (protocol level), and on the business process level.

The conceptual model of WSMO Discovery is provided in [KLe04]. In comparison with other frameworks for service discovery, WSMO Discovery provides a complete framework for discovery. Three major steps in discovery are distinguished: Goal Discovery, Web service Discovery and Service Discovery. The first step, *Goal Discovery* is about discovering abstract goal descriptions given the concrete user request. The second step, *Web service Discovery* is about how to find abstract Web service descriptions given the previous found abstract goal. The last step, *Service Discovery* is about finding real services whose abstract descriptions where discovered in the previous step. Within Web service Discovery step three principle approaches are considered: *Syntactical approaches*, *Lightweight semantic approaches* and *Heavyweight semantic approaches*.

(1) Syntactical approaches include: keyword-based search, natural language processing techniques, controlled vocabularies. *(2) Lightweight semantic* approaches include: ontologies, Action-Object-Modelling, Coarse-grained semantic description of a service. *(3) Heavyweight semantic* approaches imply that service capability is described in detail and states are took into account. A special attention in WSMO is given to the relation between discovery and mediation. This relation is more than natural when we think about the heterogeneity of the environment with different users and services using different terminologies.

In order to make communication possible between different parties mediation is required. WSMO proposes a discovery mechanism with strong mediation support. In the matchmaking process, WSMO Discovery distinguishes between four types of matchmaking: *(1) Exact Match*: all relevant services and at the same time no irrelevant services are considered; *(2) Plug-in Match*: all relevant services but also services which are considered as irrelevant for the goal can be delivered; *(3) Subsumption Match*: only relevant services but not necessary all of them are considered; and *(4) Intersection Match*: in this case of matching, the service, whose description matches the request description, is able to deliver some relevant objects, but might deliver objects which are considered as irrelevant for the goal too.

- **Query Language and Advertising Language**
  In order to express user requests and service descriptions, a family of representation languages, is provided for WSMO. This family of languages, called WSML [dBLF04], provides a formal syntax and semantics for WSMO. WSML languages: WSML-Core, WSML-DL, WSML-Flight, WMSL-Rule, WSML-Full are based on different logical formalisms: Description Logic Programming for WSML-Core, Description Logics for WSML-DL, Logic Programming for WSML-Flight and WSML-Rule, and First-Order Logic for WSML-Full. Description of user requests and service advertisements are done using concepts from ontologies that can be as well formulated in WSML.

- **Scalability**
  The WSMO approach for discovery is a complete solution that addresses a wide spectrum of approaches starting from natural language descriptions of Web services and discovery requests to precise logical definitions of Web services and requests descriptions. The actual implementation of the discovery framework proposed in

WSMO is work in progress. Scalability issues will be addressed within this work.

- **Reasoning support**
  The reasoning support required during WSMO discovery process depends on the WSML language that is used to describe requests and services. As mentioned above the WSML family of representation languages is based on a wide set of logical formalisms: Description Logics Programming, Description Logics, Logic Programming and First order Logic. An appropriate reasoner for each of these logical formalisms or even better a native reasoner for WSML languages is required during matchmaking process. The WSML group is currently developing a native WSML reasoner (see [dBFK+04]).

- **Matchmaking vs. Brokering**
  The discovery framework in WSMO does not consider the notion of brokering as defined in section 3, but this can be very easily plugged-into the framework. For matchmaking different approaches are considered: exact match, plug-in match, subsumption match, intersection match.

- **Mediation Support**
  WSMO provides a strong support for mediation. Actually in WSMO mediators are top conceptal elements that solve heterogeneity problems between different terminologies, protocols or processes. In WSMO, the mediation support for discovery is considered at different levels of this process. At keywords-based level, the mediation support considers methods like: stemming and synonym recognition. At simple semantics level, mediation support considers methods like: controlled vocabularies and shallow ontologies. Finally at the rich semantic level, the mediation support considers methods like: complete logic and heavyweight ontologies.

DAML-S/OWL-S APPROACHES Many approaches for discovery using DAML-S [The02] or OWL-S [The04] have been proposed [PKPS02], [LH03]. In [PKPS02] a DAML-S semantic matching between advertisements and requests is proposed. The matching algorithm is based on subsumption reasoning in DAML+OIL [HvH01]. A service profile and a request are considered to match when all the outputs of the request goal are matched against all, or a subset of service output, and as well all the inputs of the service are matched against all, or a subset of request goal. Different degrees of matching were identified: *(1) Exact Match*: the outputs, respectively the inputs being matched are exactly the same, *(2) Plug-in Match*: the output of the service subsumes the output of the request, *(3) Subsumes Match*: the output of the request subsumes the output of the service and *(4) Fail*: no matching services were found for the request goal.

In [LH03] a different approach for discovery using DAML-S is proposed. Compared with the previous approach all the entities of the service profile are used, namely: inputs, outputs (like in the previous approach) and as well preconditions and effects. A implemented prototype based on RACER[8] is available. Different degrees of matching are consider as well: *(1) Exact Match*: in this case the advertisement A and the request R are equivalent

[8] http://www.sts.tu-harburg.de/ r.f.moeller/racer/

concepts, *(2) Plug-in Match*: in this case the request R is a sub-concept of advertisement A, *(3) Subsumes Match*: in this case the request R is a super-concept of advertisement A, *(4) Intersection Match*: in this case the intersection of request R and advertisement A is satisfiable and *(5) Disjoint Match*: none of the matches previous presented. The strength of the match is decreasing from the *Exact Match* to *Disjoint Match*. By using a Description Logic reasoning procedure to detect possible matching, this approach inherits the time consuming operation of classifying the profiles in profile hierarchy.

- **Query Language and Advertising Language**
  The DAML-S/OWL-S approaches use the DAML-S/OWL-S specifications to model and the DAML+OIL/OWL associated languages to describe the user requests and services descriptions. All these languages are based on Description Logics logical formalism. Ontologies are the backbone in all these approaches; they offer the common terminology to describe user requests and services descriptions.

- **Scalability**
  The performance and scalability of DAML-S/OWL-S approaches for discovery depend on the architecture adopted and are influence by the performance and scalability of DL reasoners.

- **Reasoning support**
  The reasoning support for the DAML-S/OWL-S discovery approaches is determined by the underling logic mechanism used to describe the knowledge, which is Description Logics. Main reasoning approaches include: subsumption reasoning, instance checking, etc.

- **Matchmaking vs. Brokering**
  In both DAML-S/OWL-S approaches mentioned above only the matchmaking perspective is considered. No future interaction involving the matching entity after the matching is provided (no brokering approach). In both approaches different degree of matching are considered: exact, plug-in, subsumption.

- **Mediation Support**
  DAML-S/OWL-S approaches for discovery dose not provide any mediation support.

5 Summary

Table 1 presents the summary of discovery approaches evaluated according to the proposed evaluation framework.

Based on the previous evaluation the most interesting features of one approach, or combination of features from different approaches are selected and proposed as suggestions that might be considered when developing a discovery mechanism.

- **Query Language and Advertising Language**
  Semantic enabled frameworks for discovery in conjunction with rich and expressive query and advertising languages are requirements for an automatic discovery

Approach	Query and Advertising Language	Scalability	Reasoning support	Matchmaking vs. Brokering	Mediation support
Unicore	AJO language: object-oriented language	Centralized broker; brokers hierarchy	None	Brokering	None
Globus	RSL: Globus Resource Specification Language	Distributed query based service discovery	None	Globus does not provide matchmaking or brokering	None
Ontology based Matchmaker	RDF, RDF schema	Centralized querying mechanism	Deductive Database support	Matchmaking and Brokering	None
WSMO	WSML family of representation languages	Envision to be scalable	Full reasoning support (FOL, LP, DL)	Matchmaking	Complete Mediation support
DAML-S/OWL-S approaches	DAML, OWL	Dependent on DL reasoners scalability	DL reasoning	Matchmaking	None

Table 1: Evaluation of discovery approaches - Summary

mechanism. From the previous surveyed approaches OMM([TDK01]), DAML-S/OWL-S ([PKPS02], [LH03]) and WSMO([RLK04], [KLe04]) consider this aspect. Asymmetric description of resources, services and requests using an ontological support is one important feature that a good discovery mechanism must consider.

- **Scalability**
  For scalability issues approaches like Globus [FK97] can be considered. An scalable architecture possible based on hierarchial federated approach or other approaches (eg. P2P systems) is an important feature that a good discovery mechanism must consider.

- **Reasoning support**
  Depending on the advertising and query language that a discovery mechanism adopts the reasoning support will be inferred. Broad reasoning support is very well addressed in approaches like: WSMO([dBLF04], [RLK04]). For set based modelling style and reasoning (Descriptions Logics [BCM$^+$03]) only, solutions provided by approaches based on DAML or OWL might be enough. Good solutions for reasoning with rules and complex descriptions are provided by WSMO and OMM.

- **Matchmaking vs. Brokering**
  For complete discovery, in Grid environments especially, both matchmaking and

brokering functionalities must be considered. From the previous surveyed approaches only OMM considers both matchmaking and brokering.

- **Mediation support**
  Mediation support during matchmaking/discovery might be considered due to the heterogeneity of descriptions, data, etc. From the previous surveyed approaches, WSMO provides a complete solution for this problem.

6 Related Work and Conclusions

Although some survey papers about Grid and Web services in general and discovery approaches in particular exist (eg. [KBM02], [LBC+04]), none of this work presents a clear set of criteria that can be used when a discovery approach is evaluated.

In [KBM02] a taxonomy of Grid resource management systems is presented, but without a special focus in discovery. Different Grid approaches are surveyed and place in a taxonomy that considers: the type of Grid system, the resource model and the scheduling characterization.

In [LBC+04] the problem of automatic discovery and composition of Web services using semantic annotation is address. In this context a summary and evaluation of current approaches in service discovery is presented and the shortcomings of these approaches were identified. Nevertheless there is no framework, or a set of criteria that can guide the evaluation process.

In this paper, we presented an evaluation framework for discovery approaches in Web services and Grid environments. Further we have exemplified the use of our framework by evaluating some of the most relevant discovery approaches in Web services and Grid areas. Based on this evaluation, a set of suggestions for the design of a discovery solution were extracted. We believe that our framework captures most of the relevant evaluation criteria and it can be easily used to identify the strengths and weaknesses of a Grid or Web services discovery approach.

References

[ACKM03] Gustav Alonso, Fabio Casati, Harumi Kuno, and Vijay Machiraju. *Web Services*. Springer, 2003.

[BAG00] R. Buyya, D. Abramsom, and J. Giddy. Nimrod/G: An Architecture for a Resource Management and Scheduling System in a Global Computational Grid. In *International Conference on High Performance Computing in Asia-Pacific Region (HPC Asia 2000). IEEE Computing Society Press, USA.*, Beijing, China, 2000.

[BCM+03] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, 2003.

[Bec03] Dave Beckett. RDF/XML Syntax Specification (Revised). Recommendation 10 February 2004, W3C, 2003.

[BFM04] J. Brooke, D. Fellows, and J. MacLaren. Resource Brokering: The EUROGRID/GRIP Approach. Working draft, University of Manchester, Oxford Road, Manchester., 2004. Available from `http://www.allhands.org.uk/2004/proceedings/papers/178.pdf`.

[BG04] Dan Brickley and Ramanathan V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema. Recommendation 10 February 2004, W3C, 2004. Available from http://www.w3.org/TR/rdf-schema/.

[BGOA04] Ziv Baida, Jaap Gordijn, Borys Omelayenko, and Hans Akkermans. A Shared Service Terminology for Online Service Provisioning. In *In Proceedings of the Sixth International Conference on Electronic Commerce (ICEC04)*, 2004.

[CFF+04] Karl Czajkowski, Donald Ferguson, Ian Foster, Jeffrey Frey, Steve Graham, Igor Sedukhin, David Snelling, Steve Tuecke, and William Vambenepe. The WS-Resource Framework, July 2004.

[CFFK01] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman. Grid Information Services for Distributed Resource Sharing. In *Proceedings of the 10th IEEE Symposion on High Performance Distributed Computing*, 2001.

[dBFK+04] Jos de Bruijn, Cristina Feier, Uwe Keller, Ruben Lara, Axel Polleres, Livia Predoiu, and Ioan Toma. WSML Reasoner Implementation. Working draft, Digital Enterprise Research Insitute (DERI), September 2004. Available from `http://www.wsmo.org/TR/d16/d16.2/v0.2/`.

[dBLF04] Jos de Bruijn, Holger Lausen, and Dieter Fensel. The WSML Family of Representation Languages. Working draft, Digital Enterprise Research Insitute (DERI), November 2004.

[FB02] Dieter Fensel and Christoph Bussler. The Web Service Modeling Framework WSMF. *Electronic Commerce Research and Applications*, 1(2):113–137, 2002.

[FK97] Ian Foster and Carl Kesselman. Globus: A Metacomputing Infrastructure Toolkit. *Intl Journal of Supercomputer Applications*, 11(2), 1997.

[FK99] Ian Foster and Carl Kesselman. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 1999.

[FKNT02] Ian Foster, Carl Kesselman, Jeffrey M. Nick, and Steven Tuecke. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, 2002.

[Graa] Jim Gray. Grid MIDDLEWARESPECTRA. 2002 C3B Consulting Ltd.

[Grab] Jim Gray. Microsoft and Grid Computing. Memo. August 2002.

[HvH01] I. Horrocks and F. van Harmelen. Reference Description of the DAML+OIL (March 2001) Ontology Markup Language. Technical report, DAML, 2001. http://www.daml.org/2001/03/reference.html.

[KBM02] Klaus Krauter, Rajkumar Buyya, and Muthucumaru Maheswaran. A taxonomy and survey of grid resource management systems for distributed computing. *Software Practice and Experience*, 32(2):135–164, 2002.

[KLe04] Uwe Keller, Ruben Lara, and Axel Polleres (editors). WSMO Discovery. Working Draft D5.1v0.1, WSMO, 2004. Available from `http://www.wsmo.org/2004/d5/D5.1/v0.1/`.

[LBC+04] R. Lara, W. Binder, I. Constantinescu, D. Fensel, U. Keller, J. Pan, M. Pistore, A. Polleres, I. Toma, P. Traverso, and M. Zaremba. Semantics for Web Service Discovery and Composition. Technical report, Knowledge Web, December 2004.

[LH03] Lei Li and Ian Horrocks. A Software Framework for Matchmaking Based on Semantic Web Technology. In *WWW'03*, Budapest, Hungary, May 2003.

[PKPS02] M. Paolucci, T. Kawamura, T.R. Payne, and K. Sycara. Semantic Matching of Web Services Capabilities. In *Proceeding of The First International Semantic Web Conference (ISWC2002)*, Sardinia, Italy, 2002.

[Pre04] Chris Preist. A Conceptual Architecture for Semantic Web Services. In *Proceedings of the International Semantic Web Conference 2004 (ISWC 2004)*, November 2004.

[RLK04] Dumitru Roman, Holger Lausen, and Uwe Keller. Web Service Modeling Ontology – Standard (WSMO-Standard). Working draft, Digital Enterprise Research Insitute (DERI), September 2004. Available from `http://www.wsmo.org/2004/d2/v1.0`.

[Rom02] M. Romberg. The UNICORE Grid Infrastructure. *Scientific Programming, Special Issue on Grid Computing*, 10(2):149–157, 2002.

[TDK01] Hongsuda Tangmunarunkit, Stefan Decker, and Carl Kesselman. Ontology-based Resource Matching in the Grid - The Grid meets the Semantic Web. *IEEE Intelligent Systems*, 16(2), May 2001.

[The02] The DAML-S Services Coalition. DAML-S: Web Service description for the Semantic Web. July 2002.

[The04] The OWL Services Coalition. OWL-S 1.1 Beta Release. Available at http://www.daml.org/services/owl-s/1.1B/, July 2004.

[WKLW98] S. Weibel, J. Kunze, C. Lagoze, and M. Wolf. Dublin Core Metadata for Resource Discovery, 1998.