DISCOVERY CHALLENGES AND AUTOMATION FOR SERVICE-BASED APPLICATIONS IN GRID

Serena Pastore
Astronomical Observatory of Padova, National Institute of Astrophysics, vicolo Osservatorio 5, 35122, Padova, Italy

Keywords: Service discovery, semantic technologies, grid technologies, web services, service discovery.

Abstract: Discovery is a necessary task; any modern distributed system must provide this for searching and finding resources in the network according some criteria. There are many solutions for providing such tool for grid and web service environments that are essentially based on a directory service as a specialized optimized database. One great challenge in such complex distributed networks is that the effective automation of process usually fails. This paper describes the discovery issue for WSDL-based applications exported in a specific grid system by analyzing different software solutions typical to grid and web service areas. The need for automation can be partially solved with the introduction of semantic technologies that may be applied to the provider-client interaction to semantically describe the resource or directly to the registry, allowing a semantic discovery for both client and provider. Several research projects are developing software tools that will be able to be used to test the efficacy of such solutions.

1 INTRODUCTION

Discovery is a necessary task that any modern distributed system must provide. Its aim is to allow both users and applications to search and find resources in the network according to some criteria. Architectural implementations used in distributed infrastructures usually offer a mixed environment of grid system (Foster, I., Kesselman, C., 2003) and web service frameworks (Cerami, E., 2002). In the context of a project that studied the porting of astrophysical applications into grid (Benacchio, L., et al., 2005), the discovery challenge has been arisen as a key element. Once deployed in grid, any software becomes a grid resource. Therefore it should be searched and found by each grid user or application throughout the distributed system. A survey of the existing discovery solutions indicates that there are many different ways to provide this tool. Most of them rely on a directory service which is a specialized database optimized for reading, browsing and searching information to be stored. Each method places different requirements on how the information can be referenced and queried. However the main challenge in this complex environment is that the automation usually fails, requiring a manual investigation. Semantic technologies (Daconta, M.C., et al., 2003) aim to solve the automation issue that is a requirement of any discovery method. The paper describes different approaches followed in allowing discovery for Java web applications exported in a grid system as a set of web services. The solutions cover the methods used by grid information systems and also the software implementation of web service standards as a complementary method. Semantics may be applied to the provider-client interaction to semantically describe the resource or to the registry process for allowing a semantic discovery. Many research projects are in the process of developing software tools that will be able to be used in order to prove the feasibility of the solution in this specific scenario.

2 DISCOVERY METHODS FOR SOFTWARE RESOURCES

Grid and web services environments use different approaches in discovery problem solving. Information is distributed, meaning that it is spread across many disseminated machines, all of which cooperate to provide the distributed system. The focus is on the methods available for discovering

Pastore S. (2007).
DISCOVERY CHALLENGES AND AUTOMATION FOR SERVICE-BASED APPLICATIONS IN GRID.
In *Proceedings of the Third International Conference on Web Information Systems and Technologies - Internet Technology*, pages 333-336
DOI: 10.5220/0001264403330336

software resources like web applications. The hosting environment is a Java Web Services framework composed of an application and service engine (Apache Tomcat, http://tomcat.apache.org plus Apache Axis, http://ws.apache.org/axis). It manages both the HTTP transport protocol used for messages exchange and the structure of the messages involved in the transaction specified by the SOAP protocol (www.w3.org/2000/xp/Group/). The whole framework is in turn deployed on a grid machine that acts as a resource provider for grid users and applications. The node is a component of a grid site which contributes to form the INFN grid (http://grid-it.cnaf.infn.it) part of the EGEE (http://www.eu-egee.org) grid infrastructure. This grid system, built on the gLite software (http://www.glite.org), is logically organized according to the EGEE structure (EGEE JR1, 2005) in Virtual Organizations (VOs), each one consists of sites that through physical machines provide grid logic functionalities.

2.1 The Grid Web Application

Distributed technologies are largely used in the astrophysical context both as interoperable web services applications and grid applications. A global framework (the Virtual Observatory or Vobs) (McDowell, J.C., 2004) has been proposed to provide a uniform and controlled access platform to generic astronomical resources or VObs resources. The use case of the web application has been developed (Volpato, A., et al., 2004) as a VObs resource. It consists of a set of Java Web Services implementing specific querying tasks (i.e. a specific selection with SQL commands) to an astronomical catalogue. The application is described by its WSDL (http://www.w3.org/TR/wsdl) interface. The WSDL document says, in XML language, what operation the service supports and how to invoke it. It gives information about the data types used (*types* element) for all exchanged messages (*message* element), the operations performed by the service (*portType* element), and the communication protocol used for these operations (*binding* element). The set of related endpoints (*service* element) are further specified, making (*port* element) the combination of binding and network address useful as an access point. Figure 1 shows a list of such services available through the web, meaning that their automatically-generated descriptions are accessible by URL. The approaches for the discovery tasks should consider the mixed environments; thus both grid solution and web service specifications based implementations have been analyzed (fig. 2).

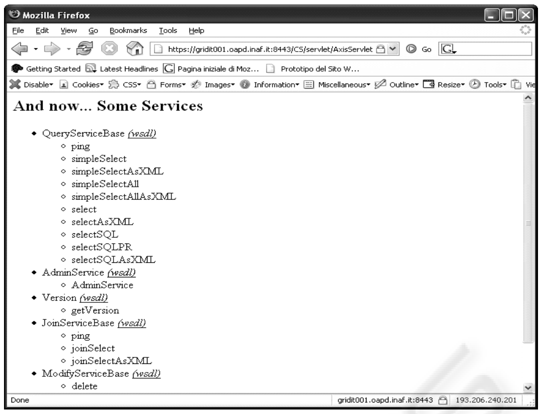

Figure 1: Examples of deployed web services accessible by a web URL.

2.2 The Different Mechanisms

Any solution in a distributed environment should provide a schema to describe the resources, a repository to store the information, a query language to interrogate them and a protocol to interact with them. Grid resources are mainly described by using the GLUE schema (http://glueschema.forge.cnaf.infn.it). This schema specifies the main features by attributes that are used as keywords in the discovery process. Usually a grid job submission includes the job's requirements in a file expressed in a specific language (Pacini, F., 2003) that uses GLUE attributes as possible values for the expressions. The method allows grid components to select a resource by performing a match between client requirements and the available resources published by the grid information system (IS). Each software resource, for example, is identified with a specific string (the `RunTimeEnvironment` attribute) representing its name; it is also associated to a grid node. The current grid IS is based on an LDAP directory service making use of the OpenLDAP software (http://www.openldap.org). It realizes (EGEE JR1, 2005) a hierarchical structure composed of a set of distributed index servers (or BDII) that maintain the list of the site's Globus MDS2 systems. The MDS2 system consists of components (GIIS/GRIS) working together to gather information coming from each node as entries in an LDAP information tree (DIT) with attributes and values. Searching tools are comprises of what is available from the software, together with some middleware toolkits. Each component of the hierarchy may be queried, and the search is based on filtering entries attributes.

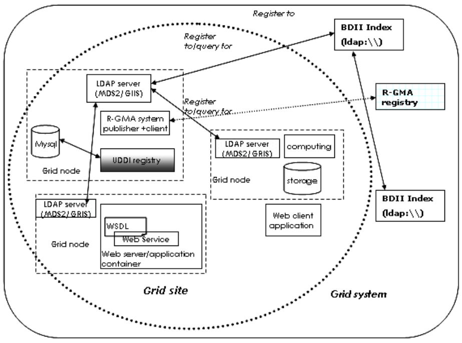

Figure 2: Different discovery components deployed in the grid system within each site.

However, the gLite toolkit is adopting the R-GMA (Relational Grid Monitoring Architecture, http://www.r-gma.org) system as its IS. Implemented as a Java web application on a site grid node, it refers to a central registry listing all the deployed systems. The method realizes a consumer-producer model and describes resources as tables in a relational database. The query language is thus based on SQL. Available searching tools consist of a browser, a command line interface that supports single query and interactive modes and other commands. The resource schema adopted is the same as in the previous solution (the R-GMA system is tagged as a software resource with the R-GMA string), but the communication protocol is different (LDAP vs. HTTP).

Web services standards, according the web services architecture that uses a consumer-producer-registry model, focus instead on a registry solution to follow the OASIS UDDI (http://www.uddi.org) specifications. By using the Apache jUDDI software (http://ws.apache.org/juddi) implementation, the registry is deployed (Pastore, S., 2005) in a grid node as a complementary approach to the discovery software resources. All information is stored as database tables like the R-GMA model, but the UDDI data model fully describes this kind of resource by specifying the provider (*businessEntity*) and its services (*businessService*), each of which is accessed via a number of bindings to protocols and physical locations (*bindingTemplate*). The UDDI objects refer to a technical models (*tModel*) structure that is a mechanism used to identify property namespace and categorization schemes. Search in UDDI is based on property-based lookup (i.e. the specific properties of a provider) or on categorization and classifications according to specific schemes (i.e. industry classification). tModels are also used as references in the mapping of WSDL features into the UDDI structure (Colgrave, J., 2004). Searching tools are web browsers and APIs, allowing operations to interact with it that use the HTTP protocol and essentially the SQL language. Table 1 summarizes the main common and differing features of the three methods. While the GLUE schema and the related methods are not sufficient to exploit software functionalities, UDDI data structures are not easily included in the grid schema. Moreover the solutions do not guarantee the automation of the discovery process.

Table 1: Summarization of main common and differing features in the three methods.

Methods	Common	Differences
BDII/MDS2 vs. R-GMA	Glue Schema	DIT/ table model; LDAP/HTTP; ldap and gLite commands/SQL
R-GMA vs. UDDI	HTTP; tables model; SQL	Glue Schema/UDDI data model

3 AUTOMATION AND SEMANTICS

All the analyzed systems require a human intervention in the process of web application discovering. Even if WSDL described capabilities and its features may be integrated into a registry, further discrimination is done by the manual inspection of the service description. The same manual activity is done using the grid discovery system. Automation challenges are partially solved with semantic technologies (http://www.w3.org/2001/sw), a set of standards and tools able to provide machine-processable descriptions of the information. Each resource is described according a semantic model in terms of classes (a set of entities), properties and relationships through a model (i.e. RDF, the Resource Description Framework) and a schema (i.e. RDF Schema), while the area of knowledge is described by an ontology through a specific language (i.e. OWL, the Web Ontology Language). In order to consider different domains (astronomical and web service knowledge), several ontologies may be combined into a single model. Studies in an astrophysical context are starting to develop an OWL-based ontology of astronomy (Shaya., E., 2006) that could better describe this area. The Semantic Web Services arm of the DAML (http://www.daml.org) program is developing a language (OWL-based web service ontology) and tools to enable the automation of services. The W3C (http://www.w3.org) has submitted a specific language called WSDL-S to associate semantic annotations with WSDL-based web services. They are the technologies applicable to the studied context. This entails two approaches:

- a client-side view adding a semantic description to the resource (client-provider interaction);
- a server-side view adding a semantic module to the registry (semantic discovery).

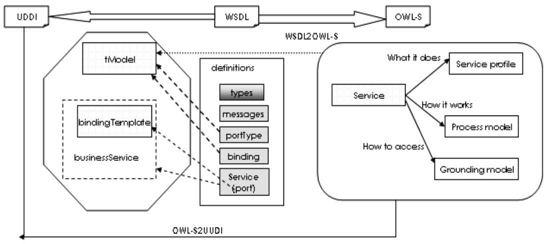

Figure 3: Relations between WSDL, UDDI and OWL-S and the available converters.

Software tools (http://projects.semwebcentral.org/) primarily developed by the Software Agents Group (http://www.cs.cmu.edu/~softagents) at Carnegie Mellon University (http://www.cmu.edu) are going to being used to test the feasibility of the various solutions. A WSDL2OWL-S converter provides a partial automatic translation between the two description languages. It is used to generate the three ontology models that make up an OWL-S document (Figure 3) and provide both the discovery information and, once found, the details needed to make use of the service. The OWL-S description of the application and its representation differs from that provided by UDDI. However, one way to combine the two efforts has been (Paolucci, M. 2002) to define a mapping between the two data structures. The mapping relates semantic models to a UDDI tModels container; it may be automatically performed by the OWL-S2UDDI software tool (Figure 3). By this conversion, OWL-S web services can be registered with UDDI. Furthermore, to exploit semantic information for the purpose of discover, UDDI engines need specific software modules added that handle semantic data (i.e. an OWL-S/UDDI Matchmaker module that allows for the processing of the OWL-S description present in the UDDI advertisement). With this approach a client discovers the agreed-upon semantic model using UDDI and loads it over standard HTTP. Then it locates the OWL document representing the semantic model by finding the appropriate tModel and accesses the service category. Having identified the relevant concepts, it navigates the mappings that link the model to the required WSDL files.

4 CONCLUSIONS

Discovery in a distributed environment merging grid systems and web service frameworks has proven to be a big challenge. The existing methods offer some characteristics in common according data schema, protocols, and tools and each method has advantages and disadvantages in addressing web application discovery. In all cases they share the same problem in providing automation. Until the introduction of semantic technologies, the best mechanism to facilitate searches will be through property-based lookup and taxonomic categorization and classification. With semantics, the web service resource can be described and thus discovered. Current research has led to semantic web services described by different languages like OWL-S and to semantic discovery which may exploit such descriptions through the use of UDDI tools. The availability of software tools that help the conversion is the basis of this feasibility study aimed at automating discovery of web software applications in a grid system.

REFERENCES

Foster, I., Kesselman, C., 2003. *The Grid 2:Blueprint for a New Computing Infrastructure*. Morgan Kaufmann.

Cerami, E., 2002. *Web Services Essentials.* O'Reilly.

Benacchio, L., et al. , 2005. In INAF Grid related activities in the framework of the Grid.it project *Workshop GRID and e-Coll. for the Space Com.*, ESA/ESRIN.

Daconta, C.M., Obrst, J.L., Kevin, T., 2003. *The Semantic Web*. Wiley Publishing.

EGEE JR1, 2005. *EGEE Middleware Architecture*. EGEE-DJRA1.1-594698-v1.0.

McDowell, J.C., 2004. *Downloading the sky*. IEEE Spectrum Online.

Volpato, A., et al., 2004. Astronomical database related applications in the Grid.it project. In *ADASSXIV, Proceedings*, Pasadena, California.

Pacini, F., 2003. *DataGrid Job Description Language Attributes*, release 2.x, DataGrid-01-TEN-00142-0_2.

Pastore, S., 2005. Searching methods for services: an UDDI solution for grid and web service environment". In *ACM Proc. of the Int. Workshop on "P2P&Service Oriented Hypermedia*", Salzburg, Austria

Colgrave, J., Januszewski, K., 2004. Using WSDL in a UDDI Registry, Ver.2.0.2. In *OASIS UDDI Spec TC.*

Shaya, E., Thomas, B., Teuben, P., Huang, Z., 2006. A Science Ontology for Goal Driven Datamining. In *Astronomy American Astr. Soc. 207th Meeting*.

Paolucci, M., Kawamura, T., Payne, T. R., Sycara, T. 2002. Importing the Semantic Web in UDDI. In *Proc of Web Services, E-business and Semantic Web Work.*