Hot Topic: Ontological Framework for a Free-Form Query Based Grid Search Engine

Chaitali Gupta
Department of Computer Science
SUNY-Binghamton
NY, USA
1-607-777-4802
cgupta1@binghamton.edu

Rajdeep Bhowmik
Department of Computer Science
SUNY-Binghamton
NY, USA
1-607-777-4802
rbhowmi1@binghamton.edu

Madhusudhan Govindaraju
Department of Computer Science
SUNY-Binghamton
NY, USA
1-607-777-4904
mgovinda@cs.binghamton.edu

ABSTRACT
If the model of free-form queries, which has proved successful for HTML based search on the Web, is made available for Grid services, it will serve as a powerful tool for scientists to retrieve information on resources, monitoring, replica location sets, and meta-data on scientific data sets, etc., in a seamless manner. To enable this vision, there is a critical need to design and develop tools that abstract away the fundamental complexity of XML based Grid specifications and toolkits, and provide an elegant, intuitive, simple, and powerful free-form query based invocation system to end users. Current implementations of XML-based Grid service descriptions require end users to have intimate knowledge of service descriptions, related toolkits, and query languages. We present our research project and initial results that employ self-learning mechanisms, matching algorithms and optimizations to match free-form user queries with corresponding operations in Grid services, and present the results to the end user. Our system uses Semantic Web concepts and Ontologies to automate discovery and matchmaking of Grid services. The research focus of this project is on the development of novel algorithms for matching user queries with correct operation names and quantifying the exact gains in accuracy due to knowledge acquisition. [1]

Categories and Subject Descriptors
D.2.2 [**Software Engineering**]: Design Tools and Techniques – modules and interfaces, user interfaces, object-oriented design methods.

General Terms
Design, Performance, Measurement.

Keywords
Grid services, Semantic Web, Ontology, Query matching.

INTRODUCTION
Grid standards such as the Open Grid Services Architecture (OGSA) and Web Services Resource Framework (WSRF) define a set of standard interfaces and behaviors of Grid services in terms of Web services based technologies. Some of the important Grid services already provide XML-based descriptions in Web service compliant interfaces. These include Information Service components, Replica Location services, Resource Management services, and many Data Grid services. For example, the MetaData Catalog Service (MCS) used by the Grid Physics Network (GriPhyN) project, runs on top of a Web service that provides functionality to store and retrieve descriptive information (meta-data) on millions of data components at varying granularity. Monitoring services provide XML interfaces to store and retrieve resource state information for tens of millions of items. Replica Location Grid Services provide meta-information on millions of files generated by scientific experiments. These services are expected to scale to larger sizes as Grids are more widely used and larger sized problems are solved in the future. The XML-based specifications provide only syntactical descriptions of the functionality provided by Grid services. Even though a wide variety of tools are available to invoke Grid services, the lack of semantics associated with XML descriptions requires user intervention in the decision making process, and in understanding the complex interfaces, contexts, composition, and invocation.

1. GRID SERVICES SEARCH ENGINE
Just as most computer users today do not have to write programs, most end users of Grid technologies should be shielded from the low-level details of these technologies. The integration of Natural Language Processing (NLP) and Information Retrieval (IR) technologies in Web search engines have made it possible for end-users to easily and effectively obtain information that is stored in billions of web pages. Users do not need professional programming expertise or technical knowledge of the structure and format in which Web pages are stored by search engine servers. As there is little context to the information that is indexed and searched via Web search engines, they typically return multiple links to the end user. However, XML based technologies and ontologies can be used to categorize and organize information in a machine-readable and understandable manner, and also to invoke remote methods to retrieve highly specific information from Grid services. Unlike Web search engines that return multiple Web pages, Grid services have the potential to return the exact information when queried using free-form text in English. Our vision is that Web semantics can be leveraged to build search engine like interfaces even for Grid services. Such a uniquely

[1] Supported in part by NSF grants IIS-0414981 and CNS-0454298.

HPDC'08, June 23–27, 2008, Boston, Massachusetts, USA.

powerful capability to use just free-form queries is not available to end-users of Grid infrastructure. It is currently required for end-users to understand query languages, have some software expertise, and knowledge of the schema formats used by various Grid services, to effectively interact with them. As a result, searching and finding resources and information is currently a challenging task. It is desirable to have a system for domain scientists to issue free-form queries such as “cluster of 32 free nodes on the OSG with at least 2GB of memory”, or “give me atmospheric data using CCSM model for 2006”, or “what is the status of job ID 117 on NYSGrid” and get back exact responses in a well structured format, such as XHTML. Developing such a framework, however, presents unique challenges.

1.1 Scope of Work

This project proposes the use of many techniques from natural language processing and semantic web to enable free-form queries. The problem of processing and acting upon arbitrary English is an extremely challenging research topic being actively addressed in the AI community. The scope of our work is therefore limited and cannot accept any free-form query. Our system is designed to accept a limited form of English with a vocabulary taken from the ontology. As the ontology will be populated using domain knowledge, we believe that free-form queries from scientists have a high likelihood of obtaining accurate results.

2. RESEARCH CHALLENGES

The research focus for this project includes the following:

• Provide access via free-form queries to a large and dynamically updated set of Grid services in various domains of interest to scientists.

• Design algorithms to automatically infer the context of user queries and map them to an appropriate set of Grid services. Section 5 presents initial results using the algorithms we have developed in this regard.

• Techniques to automatically categorize services according to the domains they cater to. The research work in this classification will leverage the work done by the Semantic Grid community in using Semantic Web concepts to describe scientific data along with some specific grid resources and services.

• An autonomous capability based on Semantic Web and Ontology technologies (OWL), WordNet [5] and YAGO [6], has been incorporated that dynamically extends and updates the ontology models to enhance the quality of results.

3. CURRENT GRID (WEB) SERVICES: SYNTAX AND SEMANTICS

SPARQL is a widely used query language for diverse data sources, including RDF formats. It is similar to SQL and can be used to query required and optional graph patterns along with their conjunctions and disjunctions. Our goal is to develop algorithms to provide the capabilities present in SPARQL via free-form queries.

Figure 1 contrasts a SPARQL query with that of a free-form query for a simple case. Suppose a user wants to check the availability of 16 nodes on the Open Science Grid (OSG) that have at least 10 gigabytes of disk space and 4GB of memory. In our system, the user needs to enter the query "names of 16 free nodes on the OSG with at least 10 Gigs and 4GB of memory”. The system will employ various algorithms to understand the query and obtain the required information by searching the appropriate XML-based services. Unlike other Grid and Web service implementations, the user does not have to fill detailed forms for each service, or use tools specific to each service. The system takes into consideration results of previous matchmaking results and utilizes it to improve performance for subsequent user queries in the same domain. It also supports memoized optimization, which use the knowledge of certain or entire parts of previously made queries, for the benefit of future queries.

SPARQL Query	**Query in the Proposed Framework**
PREFIX dc:<http://example.org/dc/element/1.1/> PREFIX ns:<http:/example.org/ns#> SELECT ?machine-name ?CPU WHERE { ?x ns:cpu ?cpu. FILTER (?cpu > 2.0). ?x dc:machine-name ?machine-name.}	“All machine names with CPU speed greater than 2.0 GHz”

Figure 1: Simple example comparing SPARQL with our Free-Form Query Based Grid Search Interface.

4. ARCHITECTURE OF THE SEMANTIC QUERY BASED FRAMEWORK FOR GRID WEB SERVICES

The architecture of our Semantic Free-Form Query framework comprises of WSDL Processor, User Query Interface, Query Processor, Lexicon, Ontology Matcher, Dictionary Matcher, and Relevance Checker. We provide a brief overview of the important modules of our system.

4.1 WSDL Processor

The WSDL processor extracts the necessary WSDL information such as operation names, part names, input output parameters, port types and service endpoints from the Grid service WSDL files.

4.2 User Query Interface

We provide a user interface similar to HTML based search engines, which accepts user queries and presents the end user with results.

4.3 Query Processor

The Query Processor processes the user query and updates its vocabulary whenever it encounters new query words. Since a user query cannot be predicted in advance, for the matching algorithms to succeed, the query words are first normalized [7].

4.4 Lexicon

The Lexicon Block is built using the WordNet 2.0 Dictionary [5]. Our system uses JWNL 1.3 API [8] to access the WordNet Dictionary. This block is used by Ontology and Dictionary Matcher, as well as Relevance Checker.

4.5 Ontology Matcher

The Ontology Matcher retrieves the ontologies from the ontology repository and matches them with the user query. We have built ontologies in OWL [9] for storing the vocabularies of major concepts like “CPU”, “memory”, “storage”, “job”, “grid job monitor service” for the initial experiments. We have used the Jena [10] framework that provides a simple OWL API for processing vocabularies and loading the ontologies into different models. The query words, fed to the Ontology Matcher, are searched in the ontology models. These models consist of statements where each statement is made up of Subject, Predicate and Object. Figure 2 shows how a simple query string ("status of jobID 117 running on NYSGrid") lights up the ontology model for a monitoring service. Within the Ontology Matcher, the Lexicon block is used and its features are employed to obtain better contextual information relevant to the client query. Initially, the query words are matched directly with the subjects, predicates, and objects of the ontology models. If no match is found, we check the ontology models with the synonyms of the query words. The four possible outcomes of this matching include - (i) neither the query word nor the synonym words are present in any of the ontology models, (ii) some of the synonyms are present, but not the query word, (iii) the query word is present, but not its synonyms, (iv) both the query word and its synonyms are present in the ontology model.

We use these outcomes to extend the ontology models, thus enriching the model. We have designed a learning module that stores the knowledge and information of a previously made query (the semantics of which are not in our ontology) to later queries for predicting more accurate results. The ontology file is extended when a synonym of a particular word yields a match. We can infer that since a synonym of the query word is present in the ontology file, the query word very likely has contextual relevance to the ontology model.

4.6 Dictionary Matcher

If the Ontology Matcher cannot determine the context of the user query, Dictionary Matcher is used to further analyze the user query. The three main techniques used in this matchmaking stage include – (i) Direct Matching, (ii) Stripped Matching, and (iii) Dictionary Level Matching. In Direct and Stripped Matching, client query words and stripped query words are matched with the actual operation names and stripped operation names respectively. The stripping is achieved by removing the stop words (get, set, of, the etc.) from the words. In Dictionary Level Matching, the synonym, hypernym (a word whose meaning denotes super class), and hyponym (a word whose meaning denotes a subordinate or subclass) of the query words are matched with the operation names and stripped operation names respectively.

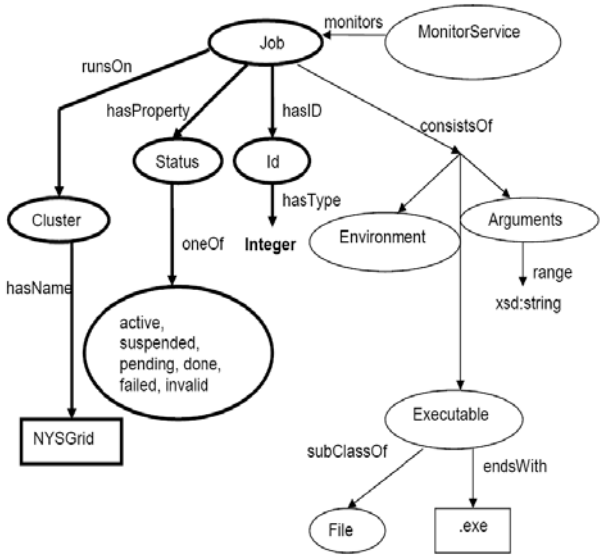

Figure 2: A query "status of job id 117 running on NYSGrid" lights up the ontology model for a Simple Job Monitoring Service Ontology.

4.7 Relevance Checker

If no match is found by the Ontology or Dictionary matcher, the system checks the glossary provided by the Lexicon as well as the input and output parameters of the methods, the part names, and the comments and annotations in the WSDL files. This aspect of the system flow is executed by the Relevance Checker.

5. PRELIMINARY FINDINGS

We conducted preliminary performance tests for the architecture described in Section 4. We used 20 sample queries of different lengths in the grid services domain. Each performance data point shown here is averaged across all ontological concepts defined in the ontology repository. We compare the performance between the domain-dependent and combined methodologies in Figure 3 when the number of elements in the ontology files is increased. The combined case considers both the domain-dependent ontologies and domain-independent methodologies. With the addition of related concepts in the OWL files, the Ontology Matcher retrieves more relevant concepts from the ontology files and as a result, the overall accuracy of the system increases from 60% to 87% for the combined case and approximately 56% to 82% for the domain-dependent case. We can note from the graph that the domain-independent methodologies are a less significant measure pertaining to the accuracy of the system. Without them, the accuracy of the system only drops by 6% on an average.

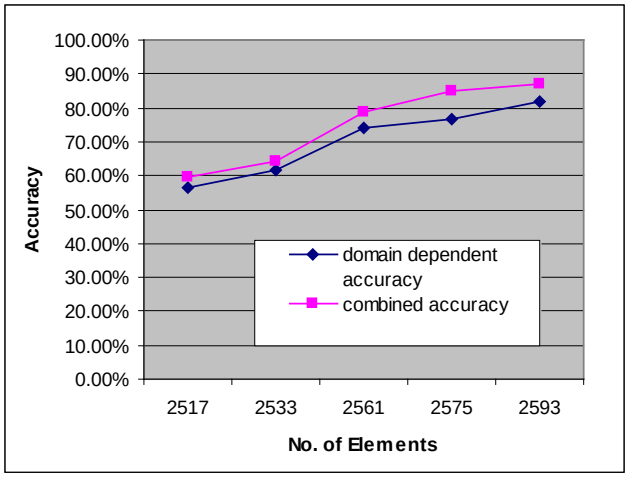

Figure 3: Accuracy with increase in the number of elements in OWL files.

6. RELATED WORK

The work that is closely related to our project includes MDS [1], Condor's Classified advertisements [2], and Gangmatching [3]. MDS provides an interface for users to query the repository for available services or resources and a matchmaker that compares the published services or resources to the requesters. Condor's ClassAds based Matchmaking framework uses a semi-structured data model that combines schema, data, and query in a simple but powerful specification language, and a clean separation of the matching and claiming phases of resource allocation. This framework is based on symmetric, attribute-based matching algorithms where the names and the values of the attributes of resources are advertised and compared with the names and values specified by the job requests. Condor's Gangmatching is designed to overcome the limitations of bilateral matching of the classified advertisement Matchmaking framework. Tangmunarunkit et al. [4] use ontology based matchmaking to the grid resource allocation and management problems, but employ Horn and F-logic based models. All these related schemes work well for scientists who have a working knowledge of the query system. Our work extends the features provided by these systems with a free-form query based interface that provides ease-of-use for domain scientists without requiring them to learn any specific XML technology or query language details.

7. REFERENCES

[1] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman, "Grid information services for distributed resource sharing", In proceedings of the Tenth IEEE International Symposium on High-Performance Distributed Computing (HPDC-10), IEEE Press, August 2001.

[2] M. Solomon R. Raman, M. Livny, "Matchmaking: distributed resource management for high throughput computing", in proceedings of the seventh IEEE International Symposium on High Performance Distributed Computing, Chicago, IL, July 1998.

[3] M. Solomon R. Raman, M. Linvy, "Resource management through multilateral matchmaking", in proceedings of the Ninth IEEE Symposium on High Performance Distributed Computing (HPDC9), pages 290–291, Pittsburgh, Pennsylvania, August 2000.

[4] Tangmunarunkit, H., Decker, S., & Kesselman, C., "Ontology-based Resource Matching in the Grid – The Grid meets the Semantic Web", in proceedings of 2nd International Semantic Web Conference, Sanibel Island, FL, USA, 2003.

[5] G. A. Miller, "WordNet: A Lexical Database for the English Language" in Comm. ACM 1983.

[6] Fabian M. Suchanek, Gjergji Kasneci, Gerhard Weikum, "Yago - A Core of Semantic Knowledge" in Proc. WWW Conference, 2007.

[7] Hai He et al., "An Automated Integrator of Web Search Interfaces for E-commerce" in VLDB Journal, Vol.13, No.3, pp.256-273, September 2004.

[8] "JWNL 1.3" Web Page: http://jwordnet.sourceforge.net/

[9] "OWL Web Ontology Language Overview" Web Page. Available: http://www.w3.org/TR/owl-features/

[10] "Jena - A Semantic Web Framework for Java" Web Page. Available: http://jena.sourceforge.net/