European Research Network on Foundations, Software Infrastructures and Applications for large scale distributed, GRID and Peer-to-Peer Technologies
A Network of Excellence funded by the European Commission

Information Services for Large-Scale Grids
A Case for a Grid Search Engine

Marios D. Dikaiakos
mdd@ucy.ac.cu
Dept. of Computer Science
University of Cyprus
1678 Nicosia, Cyprus

Rizos Sakellariou
rizos@cs.man.ac.uk
School of Computer Science
University of Manchester
Manchester, UK

Yannis Ioannidis
yannis@di.uoa.gr
Department of Informatics and Telecommunications
University of Athens, Greece

CoreGRID Technical Report
Number TR-0009
May 24, 2005

Institute on Knowledge and Data Management

CoreGRID - Network of Excellence
URL: http://www.coregrid.net

CoreGRID is a Network of Excellence funded by the European Commission under the Sixth Framework Programme

Project no. FP6-004265

Information Services for Large-Scale Grids
A Case for a Grid Search Engine

Marios D. Dikaiakos
mdd@ucy.ac.cu
Dept. of Computer Science
University of Cyprus
1678 Nicosia, Cyprus

Rizos Sakellariou
rizos@cs.man.ac.uk
School of Computer Science
University of Manchester
Manchester, UK

Yannis Ioannidis
yannis@di.uoa.gr
Department of Informatics and Telecommunications
University of Athens, Greece

CoreGRID TR-0009

May 24, 2005

Abstract

In this work we present a preliminary study of the issues surrounding the development of Seach Engines for Grid environments. We discuss the need for Grid Search Engines, that would enable the provision of a variety of Grid information services, such as locating useful resources, learning about their capabilities, and expected conditions of use. The Chapter highlights the main requirements for the design of Grid search engines and the research issues that need to be addressed.

1 Introduction

The Grid is emerging as a wide-scale, distributed computing infrastructure that promises to support resource sharing and coordinated problem solving in dynamic, multi-institutional Virtual Organisations [45]. In this dynamic and geographically dispersed setting, *Information Services* are regarded as a vital component of the Grid infrastructure [35, 55]. Information Services address the challenging problems of the discovery and ongoing monitoring of the existence and characteristics of resources, services, computations and other entities of value to the Grid. Ongoing research and development efforts within the Grid community are considering protocols, models and API's to provide an information services infrastructure that would allow efficient resource discovery and provision of information about those resources [35, 49, 55].

This research work is carried out under the FP6 Network of Excellence CoreGRID funded by the European Commission (Contract IST-2002-004265).

However, the identification of interesting and useful (in the user's context) information about the Grid can be a difficult task in the presence of too many, frequently changing, highly heterogeneous, distributed, and geographically spread resources. Equally difficult is the integration of information about different aspects of the Grid (hardware resources, data and software, policies, best practices) in order to answer complex user queries. The provision of information-services components, as currently envisaged by the Grid community, is a first step towards the efficient use of distributed resources. Nevertheless, the scale of the envisaged Grids, with thousands (or millions) of nodes, would also require well defined rules to classify the degree of relevance and interest of a given answer to a particular user. If one draws on the experience from the World Wide Web (arguably, the world's largest federated information system), efficient searching for information and services in such an environment will have to be based on advanced, sophisticated technologies that are automatic, continuous, can cope with dynamic changes, and embody a notion of relevance to a user's request. In the context of the WWW, this role is fulfilled by search engines [27].

The vision of this paper is that the technology developed as part of web search engine research, along with appropriate enhancements to cope with the increased complexity of the Grid, could be used to provide a powerful tool to Grid users in discovering the most relevant resources to requests that they formulate. Thus, our primary objective is to study issues pertaining to the development of search engines for the Grid. The remainder of this paper is organized as follows: Section 2 presents our vision for the functionality and role of Grid search engines. Section 3 surveys Grid information sources, i.e., middleware components that manage Grid-related information. In Section 4 we examine efforts to deal with the lack of standards in encoding and representing information provided by Grid information sources. Section 5 presents open problems that need to be addressed in order to build search engines for the Grid. We conclude in Section 6.

2 Vision Statement and Scenarios of Use

The Grid comprises very large numbers of heterogeneous resources distributed across multiple administrative domains (sites) and interconnected through an open network. Resources belonging to one administrative domain are usually interconnected via a local network whose performance properties (bandwidth, latency) are substantially better than those of the Internet. Coordinated sharing of resources spanning across multiple sites is made possible in the context of Virtual Organizations [45]. A Virtual Organization provides its members with access to a set of central middleware services that expose high-level functionalities, such as resource discovery, inquiry, and job submission. Through those services, the VO offers some level of resource virtualization, exposing only high-level functionalities to Grid application programmers and end-users. Additionally, the VO central services maintain and publish information about resource capabilities, supported software, available files and data-sets, etc.

In Figure 1, we provide a graphical description of the basic aspects of a VO comprising five sites interconnected via Internet. Each site makes available to the VO a range of hardware resources: supercomputers, homogeneous and heterogeneous clusters, large databases, and special devices. In addition to hardware resources, VO sites may contribute application software and services, files, and data archives. The establishment of the VO is supported by Grid middleware that runs across all sites, allowing them to export service interfaces for local resource discovery and inquiry, job submission, software invocation, file access, etc. Secure access to those service interfaces is guaranteed by the Grid security infrastructure, which deals with issues such as authentication, authorization, and access control. Grid users can request resources from and submit various types of jobs (e.g., workflows, parametric simulations) to the Virtual Organization through its central services that provide portal-based access, resource brokerage, job submission, monitoring and control.

The world-wide Grid can be considered as the collection of multiple Virtual Organizations. Different sites may use different middleware platforms (e.g., Globus and UNICORE) in order to connect their resources to the Grid. A site entering the Grid may "open" its resources to one or more VOs, with VO membership changing dynamically over time.

In such a context, we consider a search engine for the Grid as a system that facilitates the provision of a wide range of information services to Grid users, in a manner transparent to the particular characteristics of the underlying middleware. A Grid search engine is *not* intended to act as substitute to existing Grid services for resource discovery, resource inquiry or job submission on the Grid. Instead, it is expected to be a high-level entry point for users to locate useful resources, learn about their capabilities, expected conditions of use and so on, provide a unified view of resource information despite the existence of different middleware systems. This way, users can pinpoint an appropriate set of Grid resources that can be employed to achieve their goals, before proceeding with staging and submitting their job or

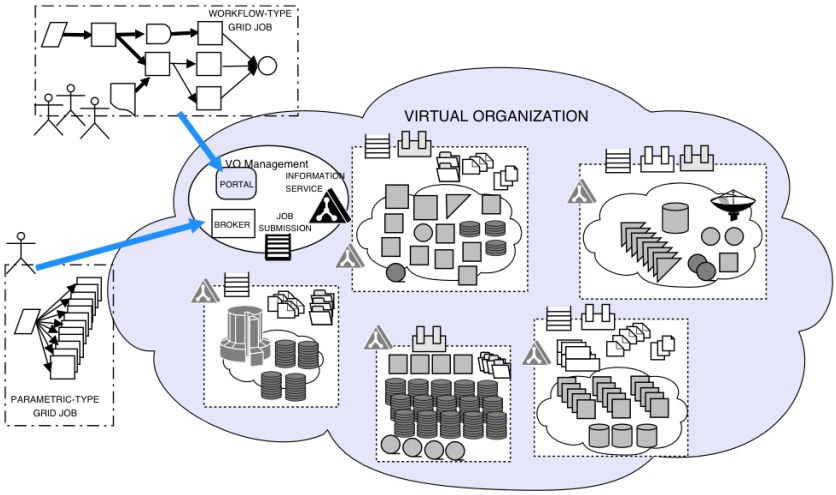

Figure 1: A view of the Grid architecture.

invoking a Grid service.

For example, a Grid search engine should be able to answer queries looking for information about:
(i) Hardware resources on the Grid, their attributes, and applicable policies of their use; for instance:

- Find an accessible Cray XT3 supercomputer with free secondary storage of at least 10 petabytes.
- Is there a VO providing exclusive access to a shared-memory multiprocessor system with at least 16 processors, 8 GB of main memory, and a usage charge of not more than 100 euros per CPU time?
- Find the access and pricing policies of a VO with Linux clusters with at least 64 dual-processor nodes connected via Myrinet.

(ii) Application services, software, and data-sets; for instance:

- Find Grid sites providing access to the LAPACK software library.
- Locate bioinformatics workflows used in AIDS research.
- Find freely available brain images encoded in DICOM.
- Find services running Quantum Chromo-Dynamics calculations (QCD) using F90 and MPI.
- Provide a list of VOs that give access to data and information about landslides.
- List the capabilities of virtual reality Grid services provided by the Foundation of the Hellenic World.
- Find workflows and datasets used in the study of the thyroid disorder.

(iii) Hardware-software combinations, Grid usage and best-practices; for instance:

- Locate Grid sites that offer access to a LAPACK software library installed on a shared-memory multiprocessor with 16 to 64 processors.
- Find the pricing and prior clientelle of Grid services that provide access to the XYZ workflow for real-time oil refinery simulations.
- Locate Grid sites that participate to the International Lattice Data Grid and have Beowulf clusters.

- Find Linux clusters with 32 to 64 dual-processor nodes on a 2-hour notice and have a documented reliability of more than 95%.
- Locate clusters with installed LAPACK libraries and a high-speed network connection to GEANT of more than 36 MBps.
- Locate Lattice-Boltzman solvers adapted for blood-flow simulations on Linux clusters and approved by the British Medical Association.
- List VOs for AIDS research that support advance reservation of resources with less than 1 hour notice and have access to Institut Pasteur XYZ database.

3 Information Sources on the Grid

Currently, a variety of Grid-middleware components collect, store, and publish collections of information that can be useful to Grid systems and users. These collections include:

- Information describing the capabilities, the operation, the status, the pricing, and the usage of *hardware resources* available on the Grid.
- Metadata about *services* deployed on the Grid, such as descriptions of functionality and interface, guidelines for invocation, and policies of use.
- Metadata regarding *data and software repositories* deployed on the Grid, describing their organization, contents, semantics, and relevant policies of access and use.
- *Job management* information regarding jobs deployed on Grids: their composition in terms of software or service components, their mapping to hardware and networking resources, their cost, etc.

The Grid middleware components that maintain such information are characterized as *Grid information* and/or *monitoring services*, although the boundaries between these two categories are not clearly defined. Typically, a Grid information service (GIS) is a *core* component of a Virtual Organization, designed to collect and provide information that is essential to the operation of the VO's infrastructure. GIS's maintain a variety of information, such as static representations of Grid-resource characteristics; descriptions of existing services, software, applicable policies, and user accounts; and dynamic representations of resource status, performance, and availability. This information is stored under a common data model and is made available to other sub-systems and end-users through a common protocol and API [34, 35, 41].

A monitoring service, on the other hand, is usually designed to monitor the status of a *specific type* of Grid resources or Grid applications. Most monitoring services are optimized to produce and process frequently changing, dynamic information collected from Grid subsystems or applications [64, 46]. Several monitoring systems, provide also filtering and statistical processing modules that produce and publish "summary" information about the monitored resources.

Grid-related information is also collected and maintained by a variety of Grid-middleware sub-systems or Grid-application components, besides GIS and monitoring services: job management information is typically maintained by resource brokers, workflow engines, logging servers, etc. Information about data repositories can be found in data-grid services, such as replica catalogues, virtual file systems, and application-specific data archives. The different information sources described above employ a variety of data models, formats, and encodings for the storage of information. Some of them, also make their data available to third-parties (i.e., to other services, administrators or end-users) by providing support for binding, discovery, and lookup through a variety of protocols and query models.

3.1 Grid Information Services

3.1.1 Globus

MDS2.x: The information services of Globus 2 [43] are provided by the *Monitoring and Discovery Service* (MDS2.x) (formely known also as Metacomputing Directory Service) [35, 41]. The goal of MDS2.x is to allow users to query for resources by name and/or by attribute, such as type, availability or load. Such queries could be of the sort of

"*Find a set of Grid nodes that have a total memory of at least 1TB and are interconnected by networks providing a bandwidth of at least 1MB/sec*" or "*Find a set of nodes that provide access to a given software package, have a certain computational capacity, and cost no more than x*," and so on. The implementation of MDS2.x is based on distributed *Directories* and the Lightweight Directory Access Protocol (LDAP) [57, 60, 63].

Under the MDS2.x approach, information about resources on the Grid is extracted by "information providers," i.e., software programs that collect and organize information from individual Grid entities. Information providers extract information either by executing local operations or contacting third-party information sources, such as the Network Weather Service [62] and SNMP. Extracted information is organized according to the LDAP data model in LDIF format and uploaded into LDAP-based servers of the Grid Resource Information Service (GRIS) [14, 60]. GRIS is a configurable framework provided by Globus for deploying core information providers and integrating new ones.

GRIS servers support the Grid Information Protocol (GRIP), an LDAP-based protocol for discovery, enquiry and communication [35]. GRIS servers can register themselves to aggregate directories, the Grid Index Information Services (GIIS). To this end, they use a soft-state registration protocol called Grid Registration Protocol (GRRP). A GIIS can reply to queries issued in GRIP. Moreover, a GIIS can register with other GIIS's, thus creating a hierarchy of aggregate directory servers. End-users can address queries to GIIS's using the GRIP protocol.

MDS3: The Information Services of Globus have been re-designed in the context of the Globus Toolkit 3 (GT3.2) release, which represents a first implementation of the Open Grid Service Architecture (OGSA) [44]. Under OGSA and Globus 3, everything is represented as a persistent or a transient *Grid service*; a Grid service is a Web service that complies to certain interface and behavioral conventions. Grid-service interfaces correspond to the portType concept of the Web Services Description Language (WSDL) and are used to manage the lifetime of Grid-service instances. Every Grid service has a particular set of associated service data, the *Service Data Elements* [8], which are represented in a standardized way.

The MDS3 component of Globus 3 is a broad framework that includes "any part of GT3 that generates, registers, indexes, aggregates, subscribes, monitors, queries, or displays Service Data Elements in some way" [7]. At the core of MDS3 lies the *Index Service*, which is one of the base services of GT3.2. The Index Service of MDS3 provides the functionality of the MDS2.x GIIS, wrapped around a Grid-service interface. In particular, the Index supports: (i) the creation and management of dynamic service data via *service-data provider programs*; (ii) the aggregation of service data from multiple Grid service instances, and (iii) the registration of multiple Grid service instances. The contents of the Index Service can be accessed by the *GT3.2 Service Data Browser*. Index Service contents can also be queried through a command-line interface that allows queries based on service-data element names or through XPath expressions.

Both MDS2 and MDS3 do not specify how entities are associated with information providers and directories, what kinds of information must be extracted from complex entities, and how different indexes can be combined into complex hierarchies. Another important issue is whether information regarding Grid entities that is stored in MDS directories or XML repositories is amenable to effective indexing. Finally, as the Grid scales to a large federation of numerous dispersed resources, resource discovery and classification become a challenging problem [49, 36]. In contrast to the Web, there is no global, distributed and simple view of the Grid's structure that could be employed to drive resource discovery and optimize replies to user queries.

3.1.2 UNICORE

The UNICORE Grid system is a set of vertically integrated software components designed to support the creation, manipulation, and control of complex batch jobs dispatched to heterogeneous systems, including supercomputers [39]. The UNICORE software architecture comprises the UNICORE client, the Gateway, the Network Job Supervisor, and the Target System Interface [39]. The definition of a UNICORE job and its resource requirements are represented as an *Abstract Job Object* (AJO), which is a collection of serialized and signed Java classes. The AJO is submitted by a UNICORE client to a selected UNICORE site through the *Gateway* component associated with that site; typically, one site comprises several Target Systems. The Gateway passes the AJO to a *Network Job Supervisor* (NJS) of a selected Target System. The NJS translates the Abstract Job into a specific batch job for the associated Target System (a process called "incarnation"); to this end, it consults its *Incarnation Database* (IDB) that contains information about Target System resources and how to run jobs on them. Furthermore, the NJS uses static information about a Target System's resources in order to make sure that the requested resources are available and comply to the TS's policies of use. A NJS can also operate as a workflow engine by passing sub-AJOs to the NJS' of peer systems. More elaborate resource brokerage is provided by the EuroGrid resource broker, which was developed in the context of the EuroGrid

project [4]. The EuroGrid broker extends NJS with a *Local Resource Checker* that checks the availability of resources on a particular site or delegates checks to broker agents at remote sites [32].

In summary, the information service functionality of UNICORE is provided in part through the Network Job Supervisor and the EuroGrid broker. UNICORE users have indirect access to this functionality through the Job Preparation Agent and the Job Monitor Controller of the UNICORE client. Information about resources and jobs, however, is neither readily available to third-party systems nor is it stored in an open format. These issues have been partly addressed in the context of the GRIP project [10, 40], which has investigated the interoperability between UNICORE and Globus. The GRIP project has developed a broker that can contact both the EuroGrid broker and Globus information services in order to locate and reserve resources for job execution across Grid infrastructures established upon Globus 2 and UNICORE. GRIP has also integrated Grid services into UNICORE [51], publishing resource information stored in NJS and IDB through Grid-service interfaces.

3.1.3 R-GMA

R-GMA is a framework that combines monitoring and information services based on the relational model [17, 34]. It has been built in the context of the EU DataGrid project and implements the Grid Monitoring Architecture (GMA) proposed by the Global Grid Forum. In brief, GMA models the information infrastructure of the Grid using three core types of components: (i) *producers* provide information; (ii) *consumers* request information; (iii) a single *registry* mediates the communication between producers and consumers [17, 64].

R-GMA implements two special properties comparing to GMA. Consumers and producers handle the registry in a transparent way; thus, anyone using R-GMA to supply or receive information does not need to know about the registry. In addition, all the information appears as one large relational database and can be queried as such. In the current implementation, the database is not distributed.

R-GMA can be used as a standalone Grid Information service, assuming information providers and consumers use the R-GMA APIs. Some tools are available to support MDS2 information providers and consumers at the expense of performance. Although the system has the potential for scalability, this remains to be demonstrated.

3.2 Grid Monitoring Systems

In order for Grid Information Services to address user's needs in locating resources of interest, they must collect information regarding the status of grid resources; this process is known as monitoring. A number of Grid monitoring systems are available to provide support with a variety of interesting to monitor entities. A detailed presentation of those systems is beyond the scope of this paper. For this purpose, we refer to a recently published taxonomy of such systems [64], which was based on the Grid Monitoring Architecture put forward by the Global Grid Forum to encourage discussion and implementations. The taxonomy is based on the system's provision of GMA components and classifies systems in four levels. The most interesting findings from this study, which are also pertinent to the scope of this paper, are that existing systems tend to have overlapping functionality, interoperability problems, while the issue of scalability does not appear to be well addressed; some preliminary work towards a scalable monitoring framework for a worldwide Grid has been presented in [65].

3.3 Grid Job Management Systems and Logging Services

The submission of a computational job to the Grid requires the description of information about that job's requirements in terms of required resources (e.g., number of processors, main memory), the location of files, etc. Currently, a number of languages for submitting computational jobs to resources exist: the Globus Resource Specification Language (RSL), Condor's ClassAds, and the EU-DataGrid Job Description Language (JDL). In order to facilitate interoperability, efforts are currently underway, by the JSDL Working Group of Grid Forum, to define an abstract standard language (the JDSL), which would encompass the common functionalities of a number of widely used batch systems [12].

Summary information about jobs submitted to and running on the Grid are collected and maintained by services, such as the Logging and Book-keeping Service (LB) developed by the European DataGrid project [6]. Logging data provide summary information about jobs submitted to the Grid. Book-keeping information is dynamic and represents the current state of a running job. The architecture of DataGrid's LB sevice comprises local deamons (*local loggers*), which are responsible for accepting messages from Resource Brokers and Job Managers via a producer API, and for passing those messages down to an *inter-logger* process. The inter-logger forwards logging messages to the

book-keeping and logging servers for storage and publication. Event messages of the LB service are encoded in the Universal Logger Message format [23]. LB information is presented as attribute-value pairs. Logging and book-keeping information describes things such as identifications of users and jobs, running jobs, input and output data, job state, and required resources.

3.4 Data and Metadata Services

Managing the vast amounts of data sets that are handled by several applications may imply the existence of some structured data management support mechanisms and/or some *metadata* or other descriptive information about the data. Different types of metadata may exist: some metadata may describe physical properties of the data objects or even of the databases that may be used; other metadata may describe content (often by means of an ontology) that allows data to be interpreted; other metadata may describe provenance. Work on data and metadata services only now begins to emerge. Ongoing work by the OGSA-DAI project [16] has led to the development of a Grid-enabled database service to provide consistent access to database metadata and to interact with databases on the Grid; this service has been used by OGSA-DQP a service-based distributed query processor for the Grid [20]. In other work, a Metadata Catalog Service to store and access descriptive metadata has been presented in [56], while a Replica Location Service for metadata information related to data replication has been presented in [33].

4 Information Modeling

The information sources described in the previous section do not follow a standard model or a common schema for organizing and representing information. Consequently, it is difficult to establish the interoperation between different Grid platforms. Moreover, the lack of common information models and standards makes it practically impossible to achieve the automated retrieval of resources, services, software, and data, and the orchestration thereof into Grid work-flows that lead to the solution of complex problems.

The need to have common, platform-independent standards for representing Grid-related information has been recognized and is currently the subject of a number of projects and working groups. These efforts have been triggered primarily by the need to enable the interoperability between large, heterogeneous infrastructures [10], and by the emergence of Open Grid Services [42, 44].

4.1 Standardizing resource information

One of the earliest efforts in that direction came from the DataTAG [2], iVDGL [11], Globus [7], and the DataGrid [6] projects, which collaborated to agree upon a uniform description of Grid resources. This effort resulted to the *Grid Laboratory Uniform Environment* (GLUE) schema, which comprises a set of information specifications for Grid resources that are expected to be discoverable and subject to monitoring [25]. GLUE represents an ontology that captures key aspects of the Grid architecture adopted by large Grid infrastructures deployed by projects like DataGrid [6], CrossGrid [5], the Large Hadron Collider Computing Grid (LCG) [13], and EGEE [3]. GLUE uses the Unified Modeling Language (UML) to describe the structure of its ontology. Information about GLUE entities is encoded in terms of named objects comprising attribute-value pairs that describe properties of the supported entities (e.g., the URI of a service, a unique ID of a resource, applicable quotas in resource use). Objects are distinguished into *auxiliary*, which carry actual information, and *structural*, which act as containers for other objects. Currently, the GLUE schema has been mapped into three different data models: LDAP, relational, and XML [24]. These models have been adopted respectively by several deployments of MDS2.x, RGM-A, and MDS3, in projects like DataGrid, LCG0, and LCG1 [21].

The GLUE ontology distinguishes two classes of entities: system resources and services that give access to system resources. Information about *system resources* is organized hierarchically and supports the following entities: clusters, sub-clusters, and nodes. *Hosts* are individual computer nodes providing processing power. *Clusters* are collections of hosts belonging to the same administrative domain (site) and linked together through a local-area network. *Sub-clusters* are sub-sets of *homogeneous* hosts that belong to the same cluster [9]. GLUE v1.1 includes also entities for physical and logical storage space: *Storage Libraries* are computers that make storage devices accessible to the Grid (disks, tapes, etc.). Storage hosted on a storage library is organized as a collection of logical *Storage Spaces*; each Storage Space has its own policies of use and access [9]. The GLUE specification includes also two *service entities*

representing the Computing and the Storage Element services of the DataGrid architecture that provide access to respective system resources. The *Computing Element* represents the entry point into a queueing system that is attached to some cluster. The *Storage Element* handles file transfers in and out of some Storage Space, using communication protocols like GridFTP [9].

Development and revisions of the GLUE schema continue in the context of the EGEE project [3], focusing primarily on issues such as the definition of an entity describing generic services, the clarification of storage resource attributes, the representation of relationships between Computing and Storage Elements, and the representation of network resources.

4.2 Standardizing job-related information

Going beyond the standardization of resources and services, a number of recent efforts are trying to devise common information representations for the structure and the status of jobs running on Grids. For example, the Job Submission Description Language Workgroup of the GGF (JSDL-WG) [12] develops the specification of the *Job Submission Description Language*, an XML Schema for describing computational batch jobs and their required execution environments. Ideally, batch jobs described with JSDL will be submitted to a computational Grid of heterogeneous batch systems that can translate to and from this abstract standard language. JSDL documents will include all information that is needed by a Grid job submission system, such as the resource requirements of a batch job, the locations of its input and output files, the techniques used for staging those files, and basic dependencies between jobs [26].

Another effort, led by the CIM Grid Schema Workgroup of the GGF [1], seeks to standardize the information that could be published by Grid schedulers about the characteristics and status of Grid jobs submitted for execution. This workgroup has adopted the Common Information Schema (CIM) of the Distributed Management Task Force's (DTMF). CIM is a conceptual information model introduced by the DTMF to facilitate the management of complex, multi-vendor, heterogeneous systems, networks, applications, and services [38]. CIM uses an object-oriented modeling approach to describe the contents and structure of an ontology of IT elements in terms of objects, classes, properties, methods, and associations. It consists of a specification that describes a modeling language and syntax for defining "real-world" managed objects (the Managed Object Format), a management schema for managed objects, a protocol that encapsulates CIM syntax and schema to provide access to those objects, and a compliance document for interoperability between vendor implementations. Based on CIM v.2.8, the GGF CIM workgroup of GGF has proposed a *Job Submission Interface Model* (JSIM) to describe the structure and attributes of batch jobs that run on Grid infrastructures [58].

Finally, the need to provide basic Grid-job accounting and resource usage information in a common format is addressed by the Usage Record (UR-WG) [18] and the Resource Usage Service (RUS-WG) [19] workgroups of the GGF. These workgroups have started working towards the proposal of XML schemas that will describe accounting information in a general, platform-independent, way.

4.3 Semantic Modeling

Because of the lack of a global schema for Grid information, several researchers are investigating the application of semantic Web technologies as an alternative for bridging the gap that exists between infrastructures with incompatible information schemas. One of the earlier efforts came from the GRid Interoperability Project (GRIP) [10]; GRIP introduces two ontologies representing the structure and attributes of UNICORE and GLUE resources, respectively. These ontologies are described in XML and fed into a tool that supports the semi-automatic association between the two ontologies. This association is used for the mapping of resource requests to hardware resources that belong to Globus and UNICORE infrastructures [31].

A similar approach for the development of an ontology-based resource matchmaker is described in [59]. The system comprises a matchmaker, which consists of three components: (i) an ontologies component, which represents the domain model and the vocabulary for expressing resource advertisements and resource requests; (ii) a domain background knowledge component containing rules that express axioms, which cannot be expressed with an ontology language; (iii) a set of matchmaking rules, which define the matching constraints between requests and resources and are expressed in a rule-based language. An ontology editor is used for the development of three domain ontologies for resources, requests, and applicable policies; these ontologies are described with the RDF-Schema specification of W3C [29]. Matchmaking is conducted with the help of a deductive database [59].

Semantic Web technologies have been proposed as a platform for the discovery of information about software and services deployed on the Grid. An early approach comes from the ICENI project in UK, which focuses on the semantic matching between Grid services and service requests in an autonomic computing context, even when requests and resources are syntactically incompatible [48]. To this end, the ICENI project proposes the concept of a *metadata space* [48]. This is an environment distinguished from the space of Grid services and resource requests. The metadata space hosts Grid-related semantic metadata, published and discovered through standard protocols. Published metadata can be classified into: (i) implementation metadata, extracted from semantic annotations of Grid services (resources); (ii) requirements metadata, which describe the semantics of resource requirements and are extracted from semantic annotations attached to resource requests; and (iii) ontologies describing the inferences that can take place during matchmaking. Semantic annotations are described in the Web Ontology Language (OWL) [54] and are attached manually to the programming interfaces of Grid-service implementation codes. The operation of the metadata space is supported by meta-services providing semantic matching and service adaptation capabilities. Service adaptation refers to the automatic adaptation of a Grid service's output to the requirements of a semantically matched but syntactically incompatible resource request. The ICENI approach is demonstrated in the case of a very simple adaptation scenario [48].

The discovery and matching of bioinformatics workflows deployed on the Grid is the goal of the myGrid project [15]. myGrid provides mechanisms for the search and discovery of pre-existing workflows based on their functionality ("task-oriented" or "construction-time" discovery), on the kind and format of their input data ("data-driven" discovery), or on the type and format of their output data ("result-driven" discovery). To make workflows discoverable, myGrid introduces the *workflow executive summary*, a workflow-specific collection of metadata represented in an XML Schema. The executive summary describes: (i) the function performed by a workflow, expressed in the terminology of myGrid's application domain (biology); (ii) the type (syntax) of a workflow's input and output data; (iii) the activities that compose a workflow and their descriptions; (iv) factual information about a workflow (its name, owner organization, location, etc.); (v) provenance information such as the workflow's author and its history [53]. Metadata belonging to the workflow executive summary include: (i) mandatory descriptions of the workflow's definition (e.g, its URI address, its script, its invocation interface, the types of its input and output data); (ii) optional *syntactic descriptions* about the format encoding of the workflow's input and output data, and (iii) optional *conceptual descriptions* of the workflow's characteristics. Workflow executive summary information is encoded in RDF with additional pointers to semantic descriptions described in OWL [54].

Two key modules in the myGrid system architecture are the *registry* and the *semantic find* component. myGrid's registry is designed to accept and store workflow descriptions, in accordance to the UDDI specification [22]. Furthermore, it supports the annotation of stored workflows with conceptual metadata [52]. myGrid's semantic find component is responsible for executing OWL queries upon the conceptual metadata attached to the workflow descriptions stored in myGrid's registry. Each time the semantic-find component receives notifications about metadata newly added to the registry, it updates accordingly an index with metadata descriptions. This index is used for fast replies to semantic queries. Alternatively, it can invoke a description-logic reasoner to answer semantic queries.

5 Discussion and Problem statement

The means used for representing and publishing resource information, in typical Grid middleware like Globus or UNICORE, do not aim to support sophisticated, user-customized queries or allow the user to decide from a number of different options. Instead, they are tied to the job submission needs within the particular environment. As we move towards a fully deployed Grid — with a massive and ever-expanding base of computing and storage nodes, network resources, and a huge corpus of available programs, services, data, and logs — providing an effective service related to the availability, the characteristics, and the usage of Grid resources can be expected to be a challenging and complex task.

As discussed earlier, efforts to address this problem are focusing on the development and standardization of information schemas (mainly defined in XML or RDF) for the description of Grid-related information. Such schemas, however, often overlap in scope and there is a clear need to re-use existing or emerging standards. Most standardization efforts, however, are still at a very early stage of development and are not adopted by new middleware systems that emerge with an increasing pace. Therefore, it is practically impossible to materialize the vision of a widely established collection of mutually compatible schemas for encoding Grid-related information. On the other hand, the use of Semantic Web technologies (ontologies, rule-based reasoning and semantic matching) faces known scalability

limitations, although it enables the resolution of complex queries upon information bases spanning across syntactically incompatible infrastructures. Finally, if we draw from the WWW experience, the identification of interesting resources has proven to be very hard in the presence of too many dynamically changing resources without well-defined rules for classifying the degree of relevance and interest of a given resource for a particular user.

Searching for information and services on the Web typically involves navigation from already known resources, browsing through Web directories that classify a part of the Web (like Yahoo!), or submitting a query to search engines [27]. In the context of the Grid, one can easily envisage scenarios where users may have to 'shop around' for solutions that satisfy their requirements best, use simultaneously different middlewares (which employ different ways to publish resource information), or consider additional information (such as, historical or statistical information) in choosing an option. The vision of this paper is that search engine technology, as has been developed for the WWW, can be used as a starting point to create a high-level interface that would add value to the capabilities provided by the underlying middleware. The integration of data discovered in and retrieved by those sources can help in the establishment and maintenance of knowledge bases for the Grid that could provide answers to various end-user queries.

5.1 Open Issues

A search engine for resource discovery on the Grid would need to address issues more complex and challenging than those dealt with on the Web. These issues are further elaborated below.

Resource Naming and Representation

The majority of searchable resources on the World-Wide Web are text-based entities (Web pages) encoded in HTML format. These entities can be identified and addressed under a common, universal naming scheme (URI). In contrast, there is a wide diversity of searchable "entities" on the Grid with different functionalities, roles, semantics, representations: hardware resources, sensors, network links, services, data repositories, software components, patterns of software composition, descriptions of programs, best practices of problem solving, people, historical data of resource usage, virtual organizations. Currently, there is no common, universal naming scheme for Grid entities.

In MDS, Grid entities are represented as instances of "object classes" following the hierarchical information schemas defined by the Grid Object Specification Language (GOS) in line with LDAP information schemas [57]. Each MDS object class is assigned an *optional* object identifier (OID) that complies to specifications of the Internet Assigned Numbers Authority, a description clause, and a list of attributes [14]. The MDS data model, however, is not powerful enough to express the different kinds of information and metadata produced by a running Grid environment, the semantic relationships between various entities of the Grid, the dynamics of Virtual Organizations, etc. Therefore, relational schemas, XML and RDF are investigated as alternative approaches for the representation of Grid entities [37, 61, 47]. Moreover, the use of a universal naming scheme, along with appropriate mapping mechanisms to interpret the resource description convention used by different middlewares, would allow a search engine for the Grid to provide high-level information services regarding resources of different independent Grids that may be based on different middlewares.

Resource Discovery and Retrieval

Web search engines rely on Web crawlers for the retrieval of resources from the World-Wide Web. Collected resources are stored in repositories and processed to extract indices used for answering user queries [27]. Typically, crawlers start from a carefully selected set of Web pages (a seed list) and try to "visit" the largest possible subset of the World-Wide Web in a given time-frame crossing administrative domains, retrieving and indexing interesting/useful resources [27, 66]. To this end, they traverse the directed graph of the World-Wide Web following edges of the graph, which correspond to hyperlinks that connect together its nodes, i.e., the Web pages. During such a traversal (crawl), a crawler employs the HTTP protocol to discover and retrieve Web resources and rudimentary metadata from Web-server hosts. Additionally, crawlers use the Domain Name Service (DNS) for domain-name resolution.

The situation is fundamentally different on the Grid: Grid entities are very diverse and can be accessed through different service protocols. Therefore, a Grid crawler following the analogy of its Web counterpart should be able to discover and lookup all Grid entities, "speaking" the corresponding protocols and transforming collected information under a common schema amenable to indexing. Clearly, an implementation of such an approach faces many complexities due to the large heterogeneity of Grid entities, the existence of many Grid platforms adopting different protocols, etc.

Definition and Management of Relationships

Web-page links represent implicit semantic relationships between interlinked Web pages. Search engines employ these relationships to improve the accuracy and relevance of their replies, especially when keyword-based searching produces very large numbers of "relevant" Web pages. To this end, search engines maintain large indices capturing the graph structure of the Web and use them to mine semantic relationships between Web resources, drive large crawls, rate retrieved resources, etc. [30, 27].

The nature of relationships between Grid entities and the representation thereof, are issues that have not been addressed in depth in the Grid literature. Organizing information about Grid resources information in hierarchical directories like MDS implies the existence of parent-child relationships. Limited extensions to these relationships are provided with cross-hierarchy links (references). However, traversing those links during query execution or indexing can be costly [55]. Alternatively, relationships can be represented through the relational models proposed to describe Grid monitoring data [37].

These approaches, however, do not provide the necessary generality, scalability and extensibility required in the context of a Grid search engine coping with user-queries upon a Grid-space with millions of diverse entities. For instance, a directory is not an ideal structure for capturing and representing the transient and dynamic relationships that arise in the Grid context. Furthermore, an MDS directory does not capture the composition patterns of software components employed in emerging Grid applications or the dependencies between software components and data-sets [28, 50]. In such cases, a Search Engine must be able to "mine" interesting relationships from monitoring data and/or metadata stored in the Grid middleware. Given that a Grid search engine is expected to be used primarily to provide summary information and hints, it should also have additional support for collecting and mining historical data, identifying patterns of use, persistent relationships, etc.

The Complexity of Queries and Query Results

The basic paradigm supported by Search Engines to locate WWW resources is based on traditional information retrieval mechanisms, i.e., keyword-based search and simple boolean expressions. This functionality is supported by indices and dictionaries created and maintained at the back-end of a search engine with the help of information retrieval techniques. Querying for Grid resources must be more powerful and flexible. To this end, we need more expressive query languages, that support compositional queries over extensible schemas [37]. Moreover, we need to employ techniques combining information-retrieval and data-mining algorithms to build proper indexes that will enable the extrapolation of semantic relationships between resources and the effective execution of user queries. Given that the expected difficulty of queries ranges from that of very small enquiries to requests requiring complicated joins, intelligent-agent interfaces are required to help users formulate queries and the search engine to compute efficiently those queries. Of equal importance is the presentation of query results within a representative conceptual context of the Grid, so that users can navigate within the complex space of query results via simple interfaces and mechanisms of low cognitive load.

6 Conclusions

The motivation for the ideas described in this paper stems from the need to provide effective information services to the users of the envisaged massive Grid. The main challenges of a Grid Search Engine, as it is envisaged, are expected to revolve around the following issues: (i) The provision of a high-level, platform-independent, user-oriented tool that can be used to retrieve a variety of Grid resource-related information in a large Grid setting, which may consist of a number of platforms possibly using different middlewares. (ii) The standardization of different approaches to view resources in the Grid and their relationships, thereby enhancing the understanding of Grids. (iii) The development of appropriate data management techniques to cope with a large diversity of information.

References

[1] CIM based Grid Schema Working Group (CGS-WG). Global Grid Forum. http://www.daasi.de/wgs/CGS/ (accessed Oct. 2004).

[2] DataTAG project. http://www.datatag.org (accessed Sept. 2004).

[3] EGEE: Enabling Grids for eScience in Europe. http://www.eu-egee.org, (accessed April 2004).

[4] EuroGrid project. http://www.eurogrid.org (accessed Sept. 2004).

[5] European CrossGrid Project. http://www.crossgrid.org (accessed April 2005).

[6] European DataGrid Project. http://www.eu-datagrid.org (accessed Sept. 2004).

[7] Globus project. http://www.globus.org.

[8] Globus Toolkit 3.2 Documentation. http://www-unix.globus.org/toolkit/docs/3.2/index.html (accessed Sept. 2004).

[9] GLUE Schema Official Documents. http://www.cnaf.infn.it/~sergio/datatag/glue (accessed March 2005).

[10] Grid Interoperability project. http://www.grid-interoperability.org (accessed Sept. 2004).

[11] iVDGL project. http://www.ivdgl.org (accessed Sept. 2004).

[12] Job Submission Description Language Working Group (JSDL-WG). Global Grid Forum. http://www.epcc.ed.ac.uk/~ali/WORK/GGF/JSDL-WG/ (accessed Oct. 2004).

[13] Large Hadron Collider Computing Grid (LCG). http://lcg.web.cern.ch (accessed Oct. 2004).

[14] MDS2.2: User Guide. http://www-unix.globus.org/toolkit/docs/3.2/infosvcs/prews/index.html (accessed Sept. 2004).

[15] myGrid UK e-Science Project. http://www.myGrid.org (accessed Nov. 2004).

[16] Open grid services architecture data access and integration (ogsa-dai). http://www.ogsadai.org.uk (accessed Dec. 2004).

[17] R-GMA: Relational Grid Monitoring Architecture. http://www.r-gma.org/ (accessed Dec. 2004).

[18] Resource Usage Service Workgroup. Global Grid Forum. https://forge.gridforum.org/projects/ur-wg/ (accessed Oct. 2004).

[19] Resource Usage Service Workgroup. Global Grid Forum. http://www.doc.ic.ac.uk/~sjn5/GGF/rus-wg.html (last accessed Oct. 2004).

[20] Service based distributed query processor (ogsa-dqp). http://www.ogsadai.org.uk/dqp (accessed Dec. 2004).

[21] The Globus GLUE Schema. http://www.globus.org/mds/glueschemalink.html (accessed Oct. 2004).

[22] Universal Description, Discovery and Integration Standard. http://www.uddi.org (accessed Nov. 2004).

[23] J. Abela and T. Debeaupuis. Universal Format for Logger Messages, May 1999. Internet Draft (expiration Nov. 1999), draft-abela-ulm-05.txt (http://www.hsc.fr/gulp/draft-abela-ulm-05.txt, accessed Oct. 2004).

[24] S. Andreozzi. GLUE Schema implementation for the LDAP data model. Technical Report Technical Report. INFN/TC-04/16, Instituto Nazionale Di Fisica Nucleare, September 2004.

[25] S. Andreozzi, M. Sgaravatto, and C. Vistoli. Sharing a conceptual model of Grid resources and services. In *Proceedings of the 2003 Conference for Computing in High Energy and Nuclear Physics*, March 2003. http://www.slac.stanford.edu/econf/C0303241/.

[26] A. Anjomshoaa, F. Brisard, A. Ly, S. McGough D. Pulsipher, and A. Savva. Job Submission Description Language (JSDL) Specification. Version 0.5.2, September 2004. http://forge.gridforum.org/projects/jsdl-wg/document/draft-ggf-jsdl-spec/en/ (accessed Oct. 2004).

[27] A. Arasu, J. Cho, H. Garcia-Molina, A. Paepcke, and S. Raghavan. Searching the Web. *ACM Transactions on Internet Technology*, 1(1):2–43, 2001.

[28] R. Bramley, K. Chiu, S. Diwan, D. Gannon, M. Govindaraju, N. Mukhi, B. Temko, and M. Yechuri. A Component based Services Architecture for Building Distributed Applications. In *Proceedings of the 9th IEEE International Symposium on High Performance Distributed Computing*, pages 51–59, 2000.

[29] D. Brickley and R.V. Guha (editors). RDF Vocabulary Description Language 1.0: RDF Schema. W3C Working Draft, October 2003. http://www.w3.org/TR/rdf-schema/.

[30] S. Brin and L. Page. The Anatomy of a Large-Scale Hypertextual (Web) Search Engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[31] J. Brooke, D. Fellows, K. Garwood, and C. Coble. Semantic matching of Grid Resource Descriptions. In M. D. Dikaiakos, editor, *Grid Computing. Second European AcrossGrids Conference, AxGrids 2004, Nicosia, Cyprus, January 2004, Revised Papers*, volume 3165 of *Lecture Notes in Computer Science*, pages 240–249. Springer, 2004.

[32] J. Brooke, D. Fellows, and J. MacLaren. Resource Brokering: The EUROGRID/GRIP Approach. In *Proceedings of the UK e-Science All Hands Meeting 2004*, 2004. http://www.allhands.org.uk/proceedings (accessed Sept. 2004).

[33] A. Chervenak, E. Deelman, I. Foster, L. Guy, W. Hoschek, A. Iamnitchi, C. Kesselman, P. Kunst, M. Ripeanu, B. Schwartzkopf, H. Stockinger, K. Stockinger, and B. Tierney. Giggle: A Framework for Constructing Scalable Replica Location Services. In *Proceedings of the 2002 ACM/IEEE conference on Supercomputing*. IEEE Computer.

[34] A. Cooke, A.J.G. Gray, L. Ma, et al. R-GMA: An Information Integration System for Grid Monitoring. In *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, volume 2888 of *Lecture Notes in Computer Science*, pages 462–481. Springer, 2003.

[35] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman. Grid Information Services for Distributed Resource Sharing. In *Proceedings 10th IEEE International Symposium on High Performance Distributed Computing (HPDC-10'01)*, pages 181–194. IEEE Computer Society, 2001.

[36] M. D. Dikaiakos, Y. Ioannidis, and R. Sakellariou. Search Engines for the Grid: A Research Agenda. In F. Rivera, M. Bubak, A. Gomez-Tato, and R. Doallo, editors, *Grid Computing. First European AcrossGrids Conference. Santiago de Compostella, Spain. February 2003. Revised papers*, volume 2970 of *Lecture Notes in Computer Science*, pages 49–58. Springer, 2004.

[37] Peter Dinda and Beth Plale. A Unified Relational Approach to Grid Information Services. Global Grid Forum, GWD-GIS-012-1, February 2001.

[38] DMTF. CIM Concepts White Paper. CIM Versions 2.4+, June 2003. http://www.dmtf.org/standards/documents/CIM/DSP0110.pdf (accessed Oct. 2004).

[39] D. W. Erwin and D. F. Snelling. UNICORE: A Grid Computing Environment. In *Lecture Notes in Computer Science*, volume 2150, pages 825–834. Springer, 2001.

[40] D. Fellows. Abstraction of Resource Broker Interface. Grid Interoperability Project. Deliverable D2.4a/UoM., November 2002. Revision 1.1.

[41] S. Fitzgerald, I. Foster, C. Kesselman, G. von Laszewski, W. Smith, and S. Tuecke. A Directory Service for Configuring High-Performance Distributed Computations. In *Proceedings of the 6th IEEE Symp. on High-Performance Distributed Computing*, pages 365–375. IEEE Computer Society, 1997.

[42] I. Foster, D. Berry, A. Djaoui, A. Grimshaw, B. Horn, H. Kishimoto, F. Maciel, A. Savva, F. Siebenlist, R. Subramanian, J. Treadwell, and J. von Reich. The Open Grid Services Architecture, Version 1.0. Open Grid Services Architecture Working Group, Global Grid Forum, July 2004. https://forge.gridforum.org/projects/ogsa-wg/.

[43] I. Foster and C. Kesselman. *The Grid: Bluepring for a Future Computing Infrastructure*, chapter Globus: A Toolkit-based Grid Architecture, pages 259–278. Morgan-Kaufmann, 1999.

[44] I. Foster, C. Kesselman, J.M. Nick, and S. Tuecke. The Physiology of the Grid. An Open Grid Services Architecture for Distributed Systems Integration. Technical report, Open Grid Service Infrastructure WG, Global Grid Forum, June 2002.

[45] I. Foster, C. Kesselman, and S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International J. Supercomputer Applications*, 15(3):200–222, 2001.

[46] M. Gerndt, R. Wismueller, Z. Balaton, G. Gombas, P. Kacsuk, Z. Nemeth, N. Podhorszki, H-L. Truong, T. Fahringer, M. Bubak, E. Laure, and T. Margalef. Performance Tools for the Grid: State of the Art and Future. APART White Paper. Technical Report Research Report Series, vol. 30, University of Technology Munich, SHAKER Verlag, 2004.

[47] D. Gunter and K. Jackson. The Applicability of RDF-Schema as a Syntax for Describing Grid Resource Metadata. Global Grid Forum, GWD-GIS-020-1, June 2001.

[48] J. Hau, W. Lee, and S. Newhouse. Autonomic Service Adaptation in ICENI using Ontological Annotation. In *Proceedings of the 4th International Workshop on Grid Computing*, pages 10–17. IEEE Computer Society, April 2003.

[49] A. Iamnitchi and I. Foster. On Fully Decentralized Resource Discovery in Grid Environments. volume 2242 of *Lecture Notes in Computer Science*, pages 51–62. 2001.

[50] C. Lee, S. Matsuoka, D. Talia, A. Sussman, M. Mueller, G. Allen, and J. Saltz. A Grid Programming Primer. Global Grid Forum, Advanced Programming Models Working Group, GWD-I, August 2001.

[51] R. Menday and P. Wieder. GRIP: The Evolution of UNICORE towards a Service-Oriented Grid. In *Proceedings of the 3rd Cracow Grid Workshop*. Academic Computer Centre CYFRONET AGH, October 2003.

[52] S. Miles, J. Papay, T. Payne, K. Decker, and L. Moreau. Towards a Protocol for the Attachment of Semantic Descriptions to Grid Services. In M. D. Dikaiakos, editor, *Grid Computing. Second European AcrossGrids Conference, AxGrids 2004, Nicosia, Cyprus, January 2004, Revised Papers*, volume 3165 of *Lecture Notes in Computer Science*, pages 240–249. Springer, 2004.

[53] S. Miles, J. Papay, C. Wroe, P. Lord, C. Goble, and L. Moreau. Semantic Description, Publication and Discovery of Workflows in myGrid. Technical Report ECSTR-IAM04-001, Electronics and Computer Science, University of Southampton, 2004.

[54] P.F. Patel-Schneider, P. Hayes, and I. Horrock. *OWL Web Ontology Language Semantics and Abstract Syntax*. World Wide Web Consortium, February 2004.

[55] B. Plale, P. Dinda, and G. von Laszewski. Key Concepts and Services of a Grid Information Service. In *Proceedings of the 15th International Conference on Parallel and Distributed Computing Systems (PDCS 2002)*, 2002.

[56] G. Singh, S. Bharathi, A. Chervenak, E. Deelman, C. Kesselman, M. Manohar, S. Patil, and L. Pearlman. A metadata catalog service for data intensive applications. In *Supercomputing 2003*, 2003.

[57] Warren Smith and Dan Gunter. Simple LDAP Schemas for Grid Monitoring. Global Grid Forum, GWD-Perf-13-1, June 2001.

[58] E. Stokes and L. Flon. Job Submission Information Model (JSIM). Version 1.0. Global Grid Forum, CIM Grid Schema Workgroup, May 2004. http://forge.gridforum.org/projects/cgs-wg.

[59] H. Tangmunarunkit, S. Decker, and C. Kesselman. Ontology-Based Resource Matching in the Grid - The Grid Meets the Semantic Web. In D. Fensel, K.P. Sycara, and J. Mylopoulos, editors, *The Semantic Web - ISWC 2003, Second International Semantic Web Conference, Sanibel Island, FL, USA, October 20-23, 2003, Proceedings*, volume 2870 of *Lecture Notes in Computer Science*, pages 706–721. 2003.

[60] G. von Laszewski and I. Foster. Usage of LDAP in Globus. http://www.globus.org/mds/globus_in_ldap.html, 2002.

[61] G. von Laszewski and P. Lane. MDSMLv1: An XML Binding to the Grid Object Specification. Global Grid Forum, GWD-GIS-002. http://www-unix.mcs.anl.gov/gridforum/gis/reports/mdsml-v1/html/.

[62] R. Wolski, N. Spring, and J. Hayes. The Network Weather Service: A Distributed Resource Performance Forecasting Service in Metacomputing. *Journal of Future Generation Computer Systems*, 15(5-6):757–768, 1999.

[63] W. Yeong, T. Howes, and S. Kille. Lightweight Directory Access Protocol. IETF, RFC 1777, 1995. http://www.ietf.org/rfc/rfc1777.txt.

[64] S. Zanikolas and R. Sakellariou. A Taxonomy of Grid Monitoring Services. *Future Generation Computer Systems*, 21(1):163–188, January 2005.

[65] Serafeim Zanikolas and Rizos Sakellariou. Towards a Monitoring Framework for Worldwide Grid Information Services. In *Proceedings of Euro-Par 2004*, volume 3149 of *Lecture Notes in Computer Science*, pages 417–422. Springer, 2004.

[66] D. Zeinalipour-Yazti and M. Dikaiakos. Design and Implementation of a Distributed Crawler and Filtering Processor. In A. Halevy and A. Gal, editors, *Proceedings of the Fifth International Workshop on Next Generation Information Technologies and Systems (NGITS 2002)*, volume 2382 of *Lecture Notes in Computer Science*, pages 58–74. Springer, June 2002.