Using the Grid and Semantic Web Technologies for Resource Management

Sabin C. BURAGA
Faculty of Computer Science,"A. I. Cuza" University of Iasi
16, Berthelot Street, RO-700483 Iasi
busaco@infoiasi.ro

***Abstract*—The paper presents a study regarding the use of the current Semantic Web technologies for knowledge management within a Grid platform. Our approach considers the existing semantic Web models, languages and technologies to be successfully applied for describing resources and services offered by a Grid system. Several examples are provided, based on important standards and initiatives, such as Resource Description Framework, Dublin Core Metadata Initiative, or Web Service Modeling Ontology.**

***Index Terms*—Distributed Computing, Grid, Knowledge Management, Semantic Web**

I. INTRODUCTION

According to [8], *Grid computing* – considered as a new paradigm for next-generation computing – enables the sharing, selection, and aggregation of world-wide distributed heterogeneous (hardware, software, logical) resources for solving large-scale problems in different areas of interest or for proving access to massive repositories of data, information, or knowledge.

Resource management and scheduling in existing environments is a complex undertaking. The geographic distribution of resources owned by diverse organization with different usage policies, cost models, and varying load and availability patterns is problematic. The producers – the owners of resources – and consumers – the users of resources – have different goals, objectives, strategies, and requirements [8].

To address these challenges, a systematic approach to model and retrieve certain resources has to be adopted. A system managing available knowledge must offer facilities for the creation, exchange, storage, and retrieval of knowledge in an exchangeable, platform-neutral and usable format [17].

The paper will describe the most important services that must be implemented within a Grid platform to accomplish the previous goals. We consider the current semantic Web models, languages and technologies can be successfully used to manage the knowledge in the context of Grid computing.

The article has the following structure: next section shortly presents the most important aspects of the Grid computing, followed by a section about the actual problems of knowledge management. Section III describes our solution of adopting the Semantic Web technologies within a Grid platform. We will provide several examples regarding the semantic annotations attached to Grid resources and the semantically enriched descriptions of Grid services via the Web Service Modeling Ontology [11].

II. GRID COMPUTING: A BRIEF PRESENTATION

The actual Internet technologies' opportunities have led to the undreamt possibility of using distributed computers as a single, unified computer resource, conducting to what is known as *Grid computing* [1, 2, 8, 18]. Grids enable the sharing, selection, and aggregation of a wide variety of heterogeneous resources, such as supercomputers, storage systems, data sources, specialized devices (e.g., wireless terminals) and others, that are geographically distributed and owned by diverse organizations for solving large-scale computational and data intensive problems in science, engineering and commerce.

One of the most used definitions is the following [22]: "Grid Computing enables virtual organizations to share geographically distributed resources as they pursue common goals, assuming the absence of central location, central control, omniscience, and an existing trust relationship".

Virtual organizations can span from small corporate departments that are in the same physical location to large groups of people from different organizations that are spread out across the globe. Virtual organizations can be large or small, static or dynamic.

A *resource* is a shared entity available in the Grid. It can be computational, such as a personal digital assistant (PDA), laptop, desktop, workstation, server, cluster, and supercomputer or a storage resource such as a hard drive in a desktop, RAID (Redundant Array of Independent Disks), and terabyte storage device. Other types of resources are the I/O ones: sensors, networks (e.g., bandwidth), printers, etc. Within a Grid, logical resources are also available: users, time counters and others.

The absence of a central location and central control implies that Grid resources do not involve a particular central location for their management. The final key point is that in a Grid environment the resources do not have prior information about each other nor do they have pre-defined security relationships [1].

Related technologies to Grid computing are peer-to-peer network architectures, cluster computing and, of course, Internet/Web computing.

Grid applications are distinguished from traditional Internet applications – mostly based on client/server model – by their simultaneous use of large number of (hardware and software) resources. That implies dynamic resource requirements, multiple administrative domains, complex and reliable communication structures, stringent performance requirements, etc. [8].

According to [1], some of the important issues regarding resource sharing across boundaries of organizations are the following:

- Identity and Authentication;
- Authorization and Policy;
- Resource Discovery;
- Resource Characterization;
- Resource Allocation;
- Resource Management;
- Accounting/Billing/Service Level Agreement (SLA);
- Security.

One of the most important initiatives concerning the Grid computing is the *Open Grid Services Architecture* (OGSA) that employs the use of Web services technologies in the context of Grid computing [2]. Grid services are in fact Web services [10] executed to give access to resources by using actual Web technologies and languages (e.g., WSDL – Web Service Description Language, SOAP protocol, XML – Extensible Markup Language) – for details, see [25].

A standardized model of infrastructure is available: *Open Grid Services Infrastructure* (OGSI). In order to include different Web Services extensions (WS-Security, WS-Trust, BPEL4WS) and to define stateful Web services, an important proposal is the Web Services Resource Framework (WSRF).

A reference implementation of the Grid architecture and Grid protocols is *Globus* providing software tools in order to build grids and Grid-based applications. These open source tools are collectively called the *Globus Toolkit* [22] – the current version is Globus 4. Other related projects are *Apple XGrid*, *Condor*, *Legion*, and *Sun Grid Engine*.

The effort of standardization of Grid protocols, architectural models, and software tools is carried by the *Global Grid Forum* [21].

III. KNOWLEDGE MANAGEMENT ISSUES

This section presents the problem of knowledge flow in the Grid-like environments. We refer to the six aspects identified in the Advanced Knowledge Technologies (AKT) objectives [15].

AKT describes six challenges that concern the engineering and management of knowledge: acquiring, modeling, reusing, retrieving, publishing and maintaining knowledge.

We focus our attention on the *Semantic Web*. According to [3] and [16], the Semantic Web is viewed as “an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation”.

When advancing towards Semantic Web, the main obstacle is the effort that the creator of information must put into organizing the knowledge and metadata (data about data), into tagging entities and relations, using vocabularies she or he must be familiar with, in order to make it comprehensible not only for humans, but also for machines. More details are given by [9], [16], and [25].

The solution for this problem is to offer software instruments which help *transparently* organizing data and metadata for machine-comprehensibility. The users of the Grid applications do not need to know semantic Web vocabularies, as the system automatically generates metadata based on the creator’s actions and on the progress of the information manipulated within the platform, in order to classify and to further retrieve given resources [12].

The system should *acquire* knowledge, and not formatted text (provided by the final users). Therefore, asking all users to markup the Website content by using XML or RDF would fail. The fundamental concept of semantic Web is the triple (*entity* has *property* with *value*). The common perception is that the entity (an information fragment), although it belongs to a structured class, is a region of an on-line document, identified by an URI (Uniform Resource Identifier). Storing the data only as formatted text causes the loss of the semantic relations between information fragments. For example, in a “Call for Papers” document it is difficult to markup the relations between the event and its location or the involved persons (organizers, invited lecturers, regular participants).

A better solution would be to store the concepts only as structured information, and display their properties/relations in the document as needed. Acquiring such structured information can be performed using (X)HTML forms. Each form field identifies the entity and the property, and the user must assign only the proper value. Another way of information gathering involves obtaining selected (semi)structured data from external sources (RSS/Atom feeds, Web services, CGI scripts, RDF stores, databases). Classes can also be defined on the basis of triple model. Each class property is specified starting from standard types and form fields are used to adapt it by defining the property name and restricting the range. For example, to model a user we must define *name* property with the restriction of maximum 50 letters as value.

Concerning *knowledge modeling*, although each piece of information should be semantically modeled, this approach seems almost impossible in practice. Our point of view is to rigorously express certain important information following the object-oriented paradigm. To properly model the knowledge, some steps must be performed [9]: the *classes* needed to be used are identified, the *properties* are defined, and the information regarding the *individuals* (class instances) is filled in via collaborative mechanisms by the involved users.

Each entity must be defined only once and attached to the mostly related document, in order to be easily reused. The information can be globally accessed within the Grid, according to certain permissions – for example, the users can edit a certain document fragment only if they belong to a given group.

Due to precise manner of structuring content, it should be easy to *retrieve* pieces of information by performing different queries. Additionally, the system will be able to automatically aggregate, retrieve, and extract information on the basis of metadata and inter-object relations. For example, ad-hoc groups of users (including the relations of collaboration between them) can be obtained from certain metadata regarding their interests.

With the help of well-known metadata standards, such as Dublin Core Metadata Initiative (DCMI) [19] and microformats [13, 23], information can be properly organized for both human and computer access and can be rendered according to the user needs and preferences. For example, the news regarding the activities of an interest group can be distributed and published in many forms by using hCalendar and RSS/Atom feeds.

IV. USING SEMANTIC GRID SERVICES

To achieve significant knowledge acquisition and management, a Grid system must support user collaborative tools and, more important, must provide a means of adding metadata (data about data) about the concepts and relations established between the resources within a given Grid platform.

A Grid system can be enhanced to support semantic Web technologies, in order to create, manage and present knowledge for any categories of users [6].

Using semantic Web-based descriptions for Grid services, the applications will automatically discover, invoke and compose the desired services. With the aid of the ontologies, the inter-operability and execution monitoring are also possible.

For example, a Grid resource accessed via a Web portal interface can embed the following metadata (in this case, different DCMI constructs about an image created during a blood-flow simulation on the Grid and supplementary annotated by two specialists):

```
<link rel="schema.DC"
   href="http://purl.org/dc/elements/1.1/" />
<meta name="DC.title" content="Patient scan"/>
<meta name="DC.description" content="This is an
image created during the blood-flow simulation."/>
<meta name="DC.creator"
   content="Blood-Flow Software Simulator"/>
<meta name="DC.contributor" content="Italo
Calvino"/>
<meta name="DC.contributor" content="Salvador
Dali"/>
<meta name="DC.created" content="2008-02-26"/>
<meta name="DC.modified" content="2008-03-27"/>
<meta name="DC.format" content="image/png"/>
<meta name="DC.language" content="en"/>
<meta name="DC.isPartOf"
   content="Blood-flow simulation"/>
```

A record about of a certain user can be expressed by the following Resource Description Framework (RDF) statement which includes the Friend Of A Friend [20] metadata:

```
<rdf:RDF xmlns:rdf=
"http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-
schema#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/">
  <foaf:PersonalProfileDocument
     rdf:about="/grid/resources/users/jsbach">
     <foaf:maker rdf:resource="#jsbach"/>
     <foaf:primaryTopic rdf:resource="#jsbach"/>
  </foaf:PersonalProfileDocument>
  <!-- an instance of the Person class -->
  <foaf:Person rdf:ID="jsbach">
     <foaf:name>Johann Sebastian Bach</foaf:name>
     <foaf:givenname>Johann</foaf:givenname>
     <foaf:family_name>Bach</foaf:family_name>
     <foaf:homepage rdf:resource=
        "http://www.music.org/bach/"/>
  </foaf:Person>
</rdf:RDF>
```

Using these annotations, any Grid resource can be easily grouped for further processing.

Additionally, Grid services can be semantically enriched by metadata and ontological descriptions. We can mention *Semantic Web Services Framework* (SWSF) which includes several proposals like *OWL-S* and *Semantic Web Services Ontology* (process ontologies for Web services) – see [24]. Another initiative is *Web Service Modeling Ontology* [11].

The following example is focused on resource management within a Grid platform. We specify a Grid service that offer the access to the metadata attached to a given resource. First, we define basic operations regarding the storage of the resources, considered as files. We can easily classify the resources by grouping them on directories (an uncomplicated taxonomy).

We must specify a simple ontology regarding the Grid resources. Using the Web Service Modeling Language, an ontology consists of *concepts* (classes), *relations* (eventually organized in hierarchies), *instances* and *axioms*.

In our example, we define only three concepts: “file”, “owner”, and “directory” – consult [4]. In addition, we define the ownership relation and an axiom which restricts an owner to be effectively member of the “owner” class:

```
ontology _"http://frontend.tuiasi.ro/gridOntology"
  nonFunctionalProperties
    dc#title hasValue "Grid ontology"
  endNonFunctionalProperties
  concept file
    name ofType _string
    hasOwner ofType owner
  concept owner subConceptOf user
    ownerOf inverseOf(hasOwner) ofType file
  concept directory
    nonFunctionalProperties
      dc#description hasValue "A directory is
uniquely identified by an i-node and includes 0 or
more items (files)"
    endNonFunctionalProperties
    inode ofType _string
    items ofType file
  relation ownership(impliesType owner,
                     impliesType file)
    nonFunctionalProperties
      dc#relation hasValue ownershipFromOwner
    endNonFunctionalProperties
  axiom ownershipFromOwner
    definedBy
      ownership(?x,?y) :-
        ?x[ownerOf hasValue ?y] memberOf owner.
  A possible instance is the following:
  instance bootstrapGlobusFile memberOf file
    name hasValue "bootstrap.jar"
    hasOwner hasValue root
```

The *dc* namespace refers to the DCMI vocabulary.

To semantically model the Grid service, we must describe its capabilities. For example, to indicate the insertion of a new file into a directory (considered as specifying a category for a given resource), we state:

```
webService
"http://frontend.tuiasi.ro/AddFileService"
    nonFunctionalProperties
      dc#title hasValue "Adding a file to a
directory"
    endNonFunctionalProperties
    importsOntology
_"http://frontend.tuiasi.ro/gridOntology"
    capability
      sharedVariables {?inode , ?filename}
    precondition
      definedBy
        ?i memberOf string and
        ?filename memberOf file.
    postcondition
```

```
    definedBy
        forall ?dir ( ?dir [ inode hasValue ?inode
]
          memberOf directory implies
          ?dir [ items hasValue ?filename])
```

The previous lines describe pre-conditions and a post-condition.

A possible interesting aspect in resource modeling and structuring is denoted by the Simple Knowledge Organization System – SKOS [14], which provides a model for expressing the basic structure and content of concept schemes (thesauri, classification schemes, taxonomies, terminologies, glossaries and other types of controlled vocabulary).

Additionally, to achieve an intelligent and automatic Grid service discovery, selection, mediation and composition into complex services, several aspects must be identified:

- Resource types – modeled by an upper-level or several domain-related ontologies;
- Syntax – textual content of resources can be represented in different syntaxes available, but the best solution is provided by the XML family of languages;
- Semantics – we must use vocabularies, metadata, ontologies, and rules;
- Transport binding – several transport mechanisms are provided by each Grid infrastructure (HTTP, HTTPS, FTP, SOAP etc.);
- Process definition/management – this aspect is related with different Web services-related standards and initiatives (e.g., WSDL, WS-Addressing, WS-Coordination, WS-Policy, and many others) [10] and process ontologies [24].

V. CONCLUSION

The paper described certain issues relating to the knowledge management within a Grid system. After a short presentation of the Grid computing, we enumerate the most important challenges that concern the engineering and management of knowledge. We presented our approach of using the Semantic Web technologies to define Grid services. We provide different examples to demonstrate the effective use of metadata and ontological constructs for specifying Grid services and the involved knowledge.

Further directions of interest are focused on using Grid technologies for disaster management [7], in conjunction to the software agents [5].

REFERENCES

[1] A. Abbas (Editor). Grid Computing: A Practical Guide to Technology and Applications, Charles River Media, 2004
[2] F. Berman, G. Fox, T. Hey (Editors). Grid Computing. Making the Global Infrastructure a Reality. Wiley, 2003
[3] T. Berners-Lee, J. Hendler, O. Lassila, "The Semantic Web", Scientific American, 5, 2001
[4] S. Buraga, "A Model for Accessing Resources of the Distributed File Systems", Advanced Environments, Tools and Applications for Cluster Computing, LNCS 2326, Springer, 2002
[5] S. Buraga, "Semantic Web Technologies in the Context of Agent Applications. From Design to Practical Deployment", Advances in Electrical and Computer Engineering, Academy of Technical Sciences of Romania, Volume 6 (13), Number 1 (25), 2006
[6] S. Buraga, L. Alboaie, "Grid Computing in the Context of Semantic Web-based Resource Management", Proceedings of the 8th International Conference on Informatics in Economy, Bucuresti, 2007
[7] S. Buraga, M. Cioca, L. Cioca, "Grid-based Decision Support System Used in Disaster Management", Studies in Informatics and Control, Number 3, 2007
[8] R. Buyya, "Economic-based Distributed Resource Management and Scheduling for Grid Computing", PhD Thesis, Monash University, Melbourne, Australia, 2002
[9] M. C. Daconta, L. J. Obrst, K. T. Smith. The Semantic Web. John Wiley & Sons, 2003
[10] T. Erl. Service-Oriented Architecture: Concepts, Technology, and Design, Prentice Hall PTR, 2005
[11] D. Fensel et al., Enabling Semantic Web Services. The Web Service Modeling Ontology, Springer, 2007
[12] B. Gupta, L. Iyer, J. Aronson, "Knowledge Management: A Taxonomy, Practices and Challenges", Industrial Management and Data Systems, 100 (1), 2000
[13] R. Khare, T., Celik, "Microformats: a Pragmatic Path to the Semantic Web", Proceedings of the 15th International Conference on World Wide Web, ACM Press, 2006
[14] A. Miles, D. Brickley (Editors), "SKOS Core Guide", W3C Working Draft, Boston, 2005: http://www.w3.org/TR/swbp-skos-core-guide
[15] N. Shadbolt, K. O'Hara (Editors), Advanced Knowledge Technologies – Selected Papers, 2004, Available: http://www.aktors.org/publications/selected-papers
[16] N. Shadbolt, W. Hall, T. Berners-Lee, "The Semantic Web Revisited". IEEE Intelligent Systems, 3(21), 2006
[17] R. Singh, L. Iyer, A. Salam, "Semantic E-business", International Journal on Semantic Web and Information Systems, 1(1), Idea Group, 2005
[18] G. Von Laszewski, P. Wagstrom, "Gestalt of the Grid", Tools and Environments for Parallel and Distributed Computing, Wiley, 2004
[19] Dublin Core Metadata Initiative, Available: http://www.dublincore.org/
[20] FOAF (Friend Of A Friend) Vocabulary Specification, Available: http://xmlns.com/foaf/0.1/
[21] Global Grid Forum, Available: http://www.globalgridforum.org/
[22] Globus, Available: http://www.globus.org/
[23] Microformats, Available: http://www.microformats.org/
[24] OWL-S, Available: http://www.daml.org/services/owl-s/1.1/overview
[25] World Wide Web Consortium, Available: http://www.w3.org/