International Journal of Web Services Research, Vol.X, No.X, 200X

Metadata, Ontologies and Information Models for Grid PSE Toolkits based on Web Services

Carmela Comito[1], Carlo Mastroianni[2] and Domenico Talia[1,2]

[1]DEIS, University of Calabria, Via P. Bucci 41 c, 87036 Rende, Italy
{ccomito, talia}@deis.unical.it

[2]ICAR-CNR, Via P. Bucci 41 c, 87036 Rende, Italy
mastroianni@icar.cnr.it

ABSTRACT:

A PSE toolkit is a group of technologies within a software architecture through which multiple PSEs can be built for different application domains. The effective use of a PSE toolkit requires the management of the heterogeneity of the involved resources that can include computers, data, network facilities, sensors, and software tools provided by different organizations. A distributed implementations of a PSE toolkit can be envisioned through the exploitation of features and functionalities offered by a service-oriented Grid framework, so obtaining a Grid PSE toolkit based on Web services. This paper presents a metadata model for Grid PSE toolkits based on Web services and the architecture of an information system that exploits the proposed metadata model. These two components contribute to define a general model of metadata management for supporting the design and implementation of problem solving environments on Grids.

KEY WORDS:
PSE, Grid, Web services, Metadata Model, Metadata Management, Information System, Ontology, WSRF.

INTRODUCTION

A problem solving environment (PSE) is a computer system that provides the computational features necessary to solve a target class of problems, according to the well-known definition reported in (Gallopoulos, Houstis et al. 1994). PSEs for industry, commercial, and business applications are gaining popularity in the recent years. An advancement of the PSE concept is the PSE toolkit concept. A PSE toolkit is a group of technologies through which multiple PSEs can be built for different application domains.

PSEs can benefit from advancements in hardware/software solutions achieved in parallel and distributed systems. In particular, the Web service paradigm and the Grid emerged as very interesting computing models in the area of parallel and distributed computing. The Web service paradigm enables flexible, platform-independent, and largely automated interactions between Web-resident services and applications, promoting the interoperation among them. The Grid is a novel infrastructure for network computing on local or geographical scales that can dynamically embody heterogeneous computing resources. Grid computing is today broadly used in many scientific and engineering application fields and is attracting a growing interest from business and industry.

The recently proposed OGSA architecture (Open Grid Services Architecture (Foster, Kesselman et al. 2002)) aligns Grid technologies with Web services technologies to take advantage of important Web services properties, such as service description and discovery, automatic generation of client and service code from service description, compatibility with emerging higher-level open standards and tools, and broad commercial support. To achieve this goal, OGSA defines a uniform exposed service semantics, the so-called Grid service, based on principles inherited from both the Grid computing and the Web services technologies.

The research and industry communities, under the guidance of the Global Grid Forum (GGF) ("GGF", 2005), contributed to evolve OGSA toward the Web Services Resource Framework (WSRF ("WSRF", 2005)) that completes the integration between Grid services and Web services. WSRF specifications define a generic and open framework for modeling and accessing stateful resources using enriched Web services referred to as WSRF Web services. This framework comprises mechanisms to describe views on the state and to support management of the state through properties associated with the Web services.

In order to fulfill the requirements of a PSE toolkit in a distributed environment, and according to the evolution trend discussed so far, this paper aims to exploit Grid and Web services features to enhance the functionalities of a PSE toolkit in a multi-domain environment. A "Grid PSE toolkit based on WSRF Web services" can indeed benefit from the advanced services and components offered by these novel technologies, such as security components, dynamic resource management services, resource discovery services, services for the parallel and distributed execution of complex applications. In particular, this paper focuses on the development of an information system for a multi-domain PSE toolkit and on the definition of a flexible and semantically enriched metadata model.

An efficient information system is a key component because a PSE toolkit needs to manage a large variety of resources that can include computers, data, network facilities, sensors, and software tools provided by different organizations (Cannataro, Folino et al 2004). The management of such heterogeneous resources requires the use of metadata that, through an accurate categorization of resources, provides useful information about the features of resources and their usage modalities.

As opposed to a single domain PSE, in a multi-domain PSE toolkit the structure of metadata information is not uniform: it depends on the type of the resource (i.e. software, hardware, data etc.), and on the application domain in which the resource is used.

Accordingly, we propose a metadata model that can be flexibly exploited in a number of application domains, and at the same time is suited to be specialized in a particular application domain. In particular, we propose to associate a metadata document to each resource offered by the PSE toolkit and distinguish three sections within that document: an ontological metadata section that identifies the resource category, a semantic metadata section that characterizes resources in different application domains and assists discovery services, and a resource metadata section that gives details about how to use and access a resource. The rationale of such distinction comes from the consideration that, in a PSE toolkit, resources must be annotated with metadata information at different levels and at different times.

Moreover, this paper introduces a novel architecture for a Grid-based information system capable to support the requirements of a multi-domain PSE toolkit. The proposed information system extends the basic information services of the WSRF-based Globus Toolkit 4 and offers semantic high-level services that exploit the proposed metadata model.

The paper is organized as follows. After the "Related Work" Section, the Section "A Metadata Model for a Grid PSE Toolkit based on Web Services" describes the metadata model and the Section "Information System of the Grid PSE Toolkit" presents a software architecture for the information system based on the proposed metadata model. The "Ontology System" Section describes the ontology system and the approach that we use to represent a domain ontology through the XML schema formalism. The Section "Conclusions" concludes the paper.

RELATED WORK

The adoption of the service oriented model in novel Grid systems, based on the OGSA architecture and the WSRF framework, has a noteworthy impact on the management of metadata and the architecture of information systems. The WSRF framework is concerned mainly with the creation, addressing, inspection, and lifetime management of stateful resources. Such a framework provides the means to manage stateful resources and codifies the relationship between Web services and stateful resources in terms of the implied resource pattern, which is a set of conventions related to the use of Web services technologies, in particular XML, WSDL, and WS-Addressing. The composition of a stateful resource and a Web service that participates in the implied resource pattern is referred to as a WS-Resource. The WSRF framework introduces the WS-Resource definition and describes how to make the properties of a WS-Resource accessible through a Web service interface. The Globus Alliance has developed the Globus Toolkit 4 (GT4) ("Globus", 2005), which offers advanced tools and functionalities based on the WSRF Web services. The information model of service-oriented Grid frameworks is essentially based on two features:

1. Metadata describing Grid services instances is stored into XML-encoded documents, called Resource Properties in WSRF. Such documents must conform to enriched XML schema documents.
2. Information is collected and indexed by means of hierarchical information services (called IndexServices in WSRF) that collect the metadata stored in WSRF Web services, aggregate it and provide enriched metadata information to high level browsing and querying services.

When exploiting the WSRF architecture, based on the Web Service technology ("WSRF", 2005), it is essential to integrate metadata embedded in services (i.e. information stored in the XML-based Resource Properties provided by WSRF Web Services) and metadata external to WSRF Web Services, which can be stored in distributed databases having extremely variable scope and completeness. In the information model proposed in this paper, this integration is achieved through the use of a metadata repository that stores the XML metadata documents related to the components/services provided by the PSE toolkit. This repository is maintained as the primary source of information for both metadata embedded in services and metadata external to them. Furthermore, XML metadata related to services can be retrieved from this repository and published within WSRF Web services in the form of Resource Properties.

In the Grid computing community there is an effort to define the so called Semantic Grid ("Semantic Grid", 2005), whose approach is based on the systematic description of resources through metadata and ontologies. In (Fox, 2003) the role of metadata in the context of the Semantic Grid is discussed. There, metadata is used to assist a three level programming model: the lowest level includes traditional code to implement a service; the next level uses agent technology and metadata to choose which services to use; the third level (workflow) links the chosen services to solve domain specific problems. The metadata model we propose aims to apply the Semantic Grid concepts in the context of a multi-domain PSE toolkit; furthermore, Semantic Grid standards – for example, the OWL-S standard – are adopted for the definition of metadata documents.

Reference (Aktas, Pierce et al 2004) describes a metadata management approach based on Semantic Web technologies, focusing particularly on the needs of the earth observation application domain. An ontology system is used to produce metadata documents in three steps. The first step aims to create a hierarchy of resource classes; then, for each class, meaningful properties are defined to characterize the resources belonging to that class. Finally, the description of classes and properties, and metadata instances, are written in semantic languages such as RDF and OWL ("OWL", 2004). The approach described in (Aktas, Pierce et al 2004) is similar to the one we propose in this paper. However, our approach is not limited to a particular application domain (such as earth observation), but can be used in multiple domains. Moreover, differently from us, (Aktas, Pierce et al 2004) does not take full advantage of the information services provided by service-oriented Grid frameworks.

In (Hastings, Langella et al 2004), a middleware framework designed for the efficient management of data and metadata in dynamic, distributed environments is described. Such a framework provides a set of services that support the distributed creation, versioning and management of metadata models and instances. XML schemas are used to represent metadata models and XML documents to represent and exchange metadata instances. In particular, users are facilitated in creating and managing XML schemas describing the data types they want to maintain, possibly using and modifying previously registered schemas. In our approach, besides using XML schemas to define the metadata model, we also exploit domain-specific ontologies (encoded in the OWL language) to enrich the semantic description of PSE resources and components and enhance the resource discovery service of the proposed information system.

A METADATA MODEL FOR A GRID PSE TOOLKIT BASED ON WEB SERVICES

In a Grid-based PSE toolkit, metadata must be used to manage heterogeneity that comes from the large variety of resources available within each resource class (Cannataro, Folino et al 2004). As compared to a PSE designed for a single application domain, a PSE toolkit covering multiple domains must tackle a further difficulty: the structure of metadata information is not uniform but depends on the characteristics of the resource under consideration. Resources can be distinguished according to their type (e.g., software, data source, hardware etc.) and the application domain in which they are used (e.g. bioinformatics, earth observation, physics etc.). In the following, the combination of a resource type and an application domain will be referred to as a resource category. In other words, a resource category is a set of resources of a given type which can be used in a given application domain.

The following resource types can be identified:

- Data-related resources, such as data sources (e.g., flat files, databases, etc), data sets (results of applications), and data management components (e.g., DBMS, file systems).
- Software resources, among which Web and Grid services are gaining a major role.
- Hosts and hardware devices (computers, storage facilities, network connections).
- Applications modeled as workflows.

The metadata model we propose takes into account the specific characteristics of different resource types and application domains. In particular, we propose to associate a metadata document to each resource offered by the PSE toolkit and distinguish three sections within that document. The rationale of such distinction comes from the consideration that, in a PSE toolkit, resources must be annotated with metadata information at different levels and at different times.

First of all, when a resource is published it is necessary to specify the category to which the resource belongs, in order to determine the set of users that could be interested in that resource: category specification is performed through the first section of metadata information. Furthermore, a resource should be semantically classified within its category to facilitate key services such as resource discovery and workflow composition: the second metadata section is used for this purpose. The third metadata section contains information that, once a resource has been discovered and selected, can be used to facilitate its access and use.

More specifically, a metadata document associated to a resource is composed of the following three sections:

1. Ontological metadata used to identify the categories to which the resource belongs. Whereas the type of a resource is univocally determined, the same resource could be used in different application domains: in such a case, ontological metadata specifies multiple categories. Ontological metadata is generated and managed by an ontology system, which also specifies the structure (expressed as an XML schema) of the remaining two sections of the metadata document. This way, it is possible to adopt a uniform approach to manage metadata information and at the same time the structure of such metadata fits the specific features of different resource categories.
2. Semantic metadata used to describe and characterize the resources belonging to a given category. To this aim, for each resource category, a set of classification parameters are defined by means of an XML schema. This schema represents the structure of semantic metadata and, as mentioned above, is constructed by an ontology system, according to the features of the application domain
3. Resource metadata supplies specific information about a resource in order to facilitate its access and usage. As well as semantic metadata, resource metadata must conform to an XML schema generated by an ontology system for each resource type. Resource metadata is further classified into description and usage metadata.

The proposed approach allows for taking advantage of the benefits offered by the Grid technology. In fact, the WSRF technology permits to store XML metadata information within a WSRF Web service, on condition that such information complies with an XML schema. This way, it is possible to exploit Grid information services to discover and access resources by examining associated metadata.

Ontological Metadata

The ontological metadata section specifies the categories to which a resource belongs (as mentioned above, a resource belongs to multiple categories if it can be used in multiple domains) and, indirectly, the XML schemas to which semantic and resource metadata must conform.

For each resource, ontological metadata is generated by an ontology system according to a high level ontology which classifies the PSE toolkit categories. Ontological metadata should specify the type of the resource (it is a service-oriented software) and the application domains in which it can be used (bioinformatics and data mining). For example, TribeMCL (Enright, Van Dongen et al. 2005) is a software tool used in the bioinformatics domain to perform data mining analysis. The ontological section of the metadata document related to TribeMCL is as follows:

```
<OntologicalMetadata>
  <ResourceType type="service">software</ResourceType>
  <ApplDomain>data mining</ApplDomain>
  <ApplDomain>bioinformatics</ApplDomain>
</OntologicalMetadata>
```

The element <ResourceType> specifies that the resource is a software tool, and it is offered as a service. Consequently, the *resource* metadata section must comply with the XML schema ServiceSoftware.xsd, which specifies the structure of resource metadata describing a generic service-oriented software.

Furthermore, the <ApplDomain> elements permit to establish that the software can be used under the data mining and bioinformatics domains. Therefore, the *semantic* metadata section must comply with the XML schemas used to categorize software in those two domains: `SciDataMiningTools.xsd`, and `BioinformaticsSoftware.xsd`. Such schemas are discussed in the Section “Ontology System”.

Semantic Metadata

Semantic metadata characterizes a resource within a given category in order to facilitate the discovery and browsing of resources. As a category identifies a couple <*resource type, application domain*>, semantic metadata describes the semantics of a resource in a specific domain. Such metadata includes parameters such as the purpose of the resource, the task achieved by the resource in that domain, the functionalities, indications about other related resources that are available in the same domain, which domain concepts the resource analyzes/describes, etc. These parameters and possible associated values are specified by means of an XML schema generated by the ontology system.

More precisely, as described in Section “Ontology System”, an ontology system offers a set of application domain ontologies. Thus, to produce semantic metadata for a given category, we will exploit the related domain ontology. Section “Ontology System” gives more details about the approach that the ontology system uses to generate an XML schema related to a category and describes the domain ontologies related to two different categories: the *scientific data mining tools* category (<*software, scientific data mining*>), and the *bioinformatics software* category (<*software, bioinformatics*>). For these two categories, the ontology system produces the XML schemas `SciDataMiningTools.xsd` and `BioinformaticSoftware.xsd`, respectively.

If a resource belongs to several resource categories (i.e., it can be used in multiple domains), the semantic metadata section is composed of as many subsections as the specified resource categories. In this case each subsection must comply with the XML schema associated to the corresponding resource category.

For example, the semantic metadata section associated to the software `TribeMCL` (see Section “Ontological Metadata”), is validated against the mentioned XML schemas `SciDataMiningTools.xsd` and `BioinformaticsSoftware.xsd` (which are shown in Section “Ontology System”). Semantic metadata, reported in Figure 1, specifies that the software analyzes BLAST protein sequences in order to predict the protein function, uses a statistical method based on the Markov Clustering algorithm (MCL), produces clusters in the form of `TribeMCL` protein families.

```
<SemanticMetadata xmlns="http://domain/path/SciDataMiningTools" …>
 <DataMiningSoftware name="TribleMCL">
  <PerformsTask>Clustering</PerformsTask>
  <ImplementsAlgorithm name="MarkovClustering" kind="ClusteringAlg">
   <UsesMethod name="MarkovModel" kind="StatisticalAnalysis">
   </UsesMethod>
  </ImplementsAlgorithm>
 </DataMiningSoftware>
</SemanticMetadata>
<SemanticMetadata xmlns="http://domain/path/BionformaticsSoftware" …>
 <BioinformaticsSoftware name="TribeMCL">
  <BiologicalFunction name="ProteinFunctionPrediction" kind="SequenceAnalisys"/>
  <BiologicalElement name="Protein" kind="BiologicalSequence"/>
  <HasInput>BLASTProteinSequence</HasInput>
  <ProducesOutput>TribeMCLProteinFamiliesSequence</ProducesOutput>
 </BioinformaticsSoftware>
</SemanticMetadata>
```

Figure 1. Semantic metadata section of the software `TribeMCL`

Resource Metadata

Resource metadata describes the procedures through which resources can be accessed and used, and can also be used to evaluate the quality of a resource. For each type of resource, the structure of resource metadata is defined through an XML schema generated by an ontology system. Such a structure does not depend on the particular application domain. Resource metadata is divided into *Description* and *Usage* metadata.

Description metadata provides a concise description of a resource. It contains provider and contact information about the entity which is responsible for providing a resource. Description metadata can also include a functional description of a resource, expressed in terms of the capabilities and functionalities offered by a resource, and information about the quality rating of such a resource. Finally, description metadata can provide information about the past usage of a resource, e.g. about the performance obtained when using the resource with given parameters and/or input data values.

Usage metadata gives information that specifies details on how to access and use a resource. Even if it would be preferable that all or most of the resources were offered as Web services, a PSE toolkit should also support non service-oriented resources. The structure of usage metadata is different for service-oriented and non service-oriented resources. Accordingly, for each type of resource (e.g. software, workflow etc.), two different XML schemas are defined. The usage metadata section of a service-oriented resource contains a reference to the WSDL document which specifies the service interface (i.e. the format of inputs and outputs), along with the URL of the service and other information. Usage metadata related to a non service-oriented resource provides detailed XML information about the resource interface, e.g. about the correct usage of a command line interface or an Application Program Interface.

In the following, for three important types of resources (i.e. software components, data resources and workflows), the structure of resource metadata is briefly outlined. We extensively exploit standards that are commonly used for such resources, and in the cases in which those standards are not sufficient, we propose additional formalisms: see reference (Mastroianni, Talia et al. 2003) for more details.

Software Resources

The resource metadata section of a software resource must be validated against the XML schema `ServiceSoftware.xsd`, or against the schema `GenericSoftware.xsd`, depending on whether the software is offered as a service or not. The two cases are discussed separately in the following.

Service-oriented software. The *usage* metadata section must contain at least a reference to the WSDL document describing the service. However, the WSDL language does not give semantic information about a Web service due to the limited expressive power of the XML Schema formalism. In the recent years, a number of formalisms have been proposed to describe the semantics of a service. One important proposal has been formulated by the DARPA Agent Markup Language (DAML) Program (“OWL-S”, 2005). The Semantic Web Services arm of the DAML program developed an OWL-based Web Service Ontology, namely OWL-S, to enable automation of services on the Semantic Web. An OWL-S document gives different types of semantic information about a Web service, through the definition of the following OWL classes:

- the *Profile* class, which gives information about the service provider and a functional description of the service;
- the *Model* class, which describes the internal process that realizes the service;
- the *Grounding* class, that specifies details about the access mechanisms.

If a description of a service is furnished through the OWL-S language, the *description* subsection should contain a reference to that description. WSDL and OWL-S documents can also reference each other.

Non service-oriented software. Resource metadata should provide the same type of information which, in the case of service-oriented resources, is provided by OWL-S and WSDL documents: structure of input and output, information about the software provider, functional description etc. Such information is contained in an XML document. Details on the syntactic description of a software interface are given in (Mastroianni, Talia et al. 2003).

Data Resources

Data-related resources can be classified as follows:

1. *Data resource managers* are systems designed to manage data. Examples are a file system or a DBMS.
2. *Data sources* can be files, relational databases, XML databases, transaction databases, etc.
3. *Data sets* are collections of data that are not explicitly managed by a resource manager. For example, data generated by an application or the result set of a query evaluated over a relational database.

Resource metadata for data-related resources is validated against the XML schema `ServiceData.xsd`, or against the schema `GenericData.xsd`, depending on whether the resource is offered as a service or not. The two cases are discussed separately below.

Non service-oriented data resources. Description metadata includes:

- *Product information metadata* defining technical parameters such as product name, data currency and history (i.e. versions).
- *Structure metadata.* It contains information about both the logical/physical structure of a data source (e.g. organization and grouping of data items into logical records, database schemas etc.) and the data model.
- *Capability metadata* specifies the capabilities of a data resource manager. For a DBMS such metadata specifies: language capabilities, queries and update operations supported; transactional capabilities; connection options such as protocols and encodings that can be supported, etc.

Service-oriented data resources. We adopt the *Open Grid Services Architecture Data Access and Integration* ("OGSA-DAI", 2005) standard. It builds upon OGSA data access components to manage both relational and XML databases wrapped as Grid Data Services (GDSs). Metadata is handled through several types of XML documents including: (i) a data resource configuration document specifying the activities that a GDS can support, information on the database management system, on the connection to data resources, etc; (ii) a `RoleMap` file containing data sources access permissions; (iii) a registry containing information about a set of GDSs; (iv) a `gridDataServicePerform` document used by clients to send query and update operations to a GDS. (v) a `gridDataServiceResponse` document, returned by a GDS, which contains the results of query and update operations.

Workflows

A main purpose of a Grid-based PSE toolkit is to facilitate users in the specification of complex applications and in the construction of workflows composed by multiple tasks that must be executed sequentially or in parallel.

Several Grid-based workflow systems, such as Pegasus (Deelman, Blythe et al. 2003)), adopt a two layer approach to build and execute a workflow: an abstract workflow is designed at a high level, and then is mapped to the set of available Grid resources, thus generating an executable workflow. In our system, a *concrete workflow* contains only well defined resources (e.g. particular software resources to be executed on specified hosts), whereas an *abstract workflow* contains at least an *abstract resource*, that is a resource defined by means of constraints on metadata properties (e.g., a software that extracts clusters from bioinformatics data). The instantiation of an abstract workflow resolves each abstract resource into a concrete resource available on the Grid.

The document that describes a concrete workflow is placed in the *resource* section of the metadata document describing the application. Details about the specification of abstract and concrete workflows with XML syntax, and about workflow instantiation, are given in (Mastroianni, Talia et al. 2003).

If an application is composed of Web services, a concrete workflow can also be expressed with one of the formalisms that are emerging for this purpose, such as OWL-S ("OWL-S", 2005) and BPEL (Curbera, Goland et al. 2005). An OWL-S *composite process* can be viewed as a workflow that maintains and manages an internal state; each message the client sends advances the state through the workflow. BPEL defines a model and a grammar for describing the behaviour of a business process based on interactions between the process and its partners.

INFORMATION SYSTEM OF THE GRID PSE TOOLKIT

To properly manage metadata in a Grid PSE toolkit based on Web services, we model metadata on the basis of the above described approach and propose an information system that accomplishes two main tasks: managing metadata and supporting high-level discovery services. Figure 2 depicts the architecture of the information system.

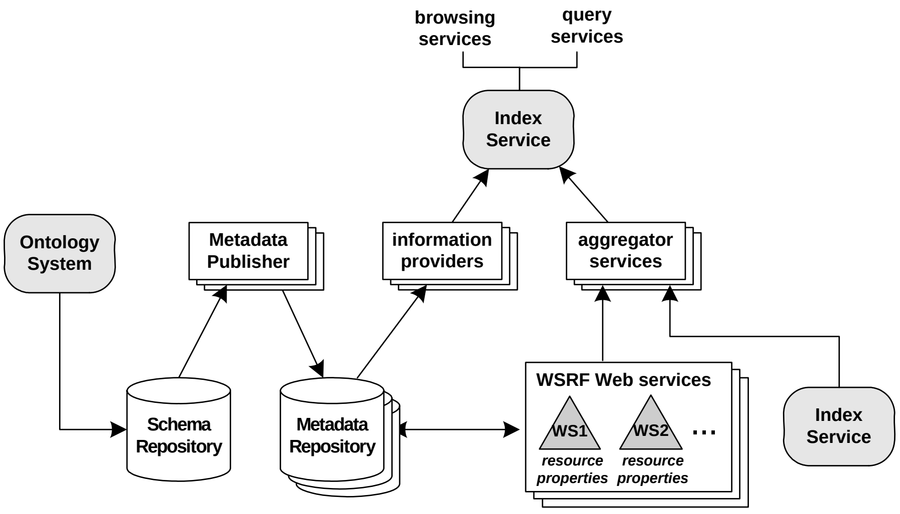

Figure 2. Architecture of the PSE toolkit information system

The information system is integrated with the WSRF-based Globus Toolkit 4 ("Globus", 2005), in order to take advantage of the services offered by that framework (browsing and indexing services, information providers etc.).

The information system is composed both by fully distributed components and hierarchical components. In particular, components that are used to manage, publish and access metadata documents are distributed on the different hosts. The schema repository, that stores the XML schemas generated by the ontology system, and the metadata repository, that stores the XML metadata documents, are both distributed XML databases. The ontology system and the components that are used to index, browse and search resources on the Grid are organized in a hierarchical configuration that reflects the structure of Grid Virtual Organizations. In Figure 2, components that are inherently distributed are replicated.

The rest of the section is organized as follows. Section "Metadata Repository and WSRF Web Services" explains the approach used to store metadata documents and justifies the opportunity of storing metadata both in the metadata repository and within WSRF Web services. Section "Index Services" describes the main characteristics of GT4 Index Services. Finally, Section "Publishing and Discovery of Resources" illustrates the publishing and discovering functionalities offered by the proposed information system. The structure of the ontology system is analyzed in more details in Section "Ontology System".

Metadata Repository and WSRF Web Services

The metadata repository stores the metadata documents related to the components/services provided by the PSE toolkit. As mentioned in Section "A Metadata Model for a Grid PSE Toolkit based on Web services", the choice of using XML schemas to define the structure of metadata allows for an efficient integration with the GT4 framework. In GT4, metadata is stored within Web services as WSRF *resource properties,* whose structure is defined by means of enriched XML schemas. As a consequence, a metadata document associated to a service-oriented resource, or part of such a document, can be retrieved from the metadata repository and stored within a WSRF Web service.

The advantage of storing metadata both in the metadata repository and within a service is motivated as follows. The publication of metadata within a service is useful if we want to take advantage of the Grid information services offered by the Globus Toolkit. On the other hand, storing metadata in the metadata repository is useful for two reasons: (i) to give persistency and high availability to metadata; (ii) to provide a uniform point of access to metadata, including metadata describing non service-oriented resources.

However, consistency problems could arise. To tackle this issue, the metadata repository is chosen as the primary source of information. Metadata associated to a new resource is generated by the metadata publisher and stored in the metadata repository. If the new resource is a Web service, metadata is retrieved by an *information provider* and published as resource properties. One information provider is associated to each Web service, and is executed when the service is published for the first time and whenever the metadata document stored in the repository is modified by authorized users.

It is also possible that resource properties are modified during the lifetime of a Web service. To avoid inconsistency problems, an attempt to modify a resource property requires an access to the metadata document stored in the metadata repository. If access is authorized, a lock is executed on the database, the requested modification is performed on the metadata document with a synchronous operation and finally the resource property is modified as requested.

The metadata repository adopted in the PSE toolkit is a distributed XML database based on the Apache Xindice ("Xindice", 2005) platform. For each Grid node, the metadata repository contains metadata related to all the resources published in that node. To facilitate the searching and browsing of resources, metadata can also be aggregated and published by GT4 Index Services, as described in the next subsection.

Index Services

The GT4 information system produces, aggregates and indexes metadata related to the resources provided by a set of Grid hosts belonging to a Virtual Organization (VO). Such a system exploits the functionalities of GT4 Index Services; usually each VO provides one Index Service, but more Index Services, organized in a hierarchy, can be installed on a large VO. Metadata describing the resources of the PSE toolkit is aggregated and published on Index Services with two mechanisms, depending on the kind of resource:

1. *Non service-oriented resources*. A set of information providers retrieve the XML metadata documents stored in the metadata repositories of a VO, and publish them in the Index Service of that VO.
2. *Service-oriented resources*. The VO Index Service subscribes to the resource properties that have to be aggregated and indexed, in order to be notified of changes. The GT4 *service aggregators* retrieve the resource properties from the WSRF Web services and publish them in the Index Service. If the Index Services of a VO are organized in two or more levels, service aggregators can retrieve metadata from lower level Index Services and publish it in higher level Index Services, as depicted in Figure 2.

Since Index Services are fed with data retrieved both from Web services and metadata repositories, the deployed architecture provides a uniform and flexible mechanism to query and browse metadata related to all kinds of resources, including non service-oriented ones. Browsing and querying can be performed by means of specific Globus Toolkit services (e.g. the Service Data Browser) or high level services offered by domain specific PSEs.

Publishing and Discovery of Resources

On top of GT4 Index Services, the proposed architecture provides high level services through which users can publish and discovery resources on the PSE toolkit.

The *publishing* functionality enables to create, modify, and delete XML metadata documents stored in the metadata repository and within Web services. The information system offers an assisted publishing procedure that guarantees the consistency of a metadata document with the XML schemas associated to a given resource category. In particular, when a user publishes a new resource, the following steps are performed:

1. The user verifies if the new resource belongs to one of the resource categories defined by the ontology system. It can occur that:
   (a) the resource category under consideration has already been defined. In this case, the user can exploit the ontology system to fill the ontological section of the resource metadata document (see Section “Ontological Metadata”) and retrieve the structures – i.e the XML schemas - of semantic and resource metadata sections from the schema repository (see Sections “Semantic Metadata” and “Resource Metadata”).
   (b) the new resource does not belong to any defined resource category. In this case the user, with the aid of domain experts, can use the ontology system to refine the classification of application domains, and possibly create a new resource category and the corresponding XML schemas that will be stored in the schema repository. Afterwards the user will be able to use such schemas and produce the new resource metadata document.
2. The user exploits the metadata publisher to create the semantic and resource sections of the metadata document which describes the new resource. The publisher offers a semi-automatic

tool for editing metadata documents: it allows a user to (i) view the characteristics of the resource categories defined by the ontology system and (ii) define the parameters and values through which the new resource can be described and classified.

3. At the end of the editing process, the metadata document is stored in the metadata repository.
4. If the new resource is offered as a Web service, its metadata document, or part of it, is also published within the service itself as a set of resource properties .
5. Metadata stored within the service can be aggregated and published by the GT4 Index Service.

The *Discovery* functionality allows users to search, locate and select PSE components and resources by examining the metadata information contained in each node of the Grid. To this end, the Index Service offers a set of services for browsing and querying metadata documents made available by the PSE toolkit. Typically, a user specifies a set of constraints on resource features, and the information system matches such constraints with the semantic section of the resource metadata documents.

Specifically, when a user needs to discover resources having given characteristics, she/he executes the following actions:

1. To construct a query, the user must know the XML schema that defines the structure of the semantic metadata section for the resource category under consideration. If such a schema is not known, the user can browse or query the ontology system in order to retrieve it from the schema repository.
2. The user builds the query on the basis of the parameters and possible values specified in the XML schema.

The user submits the query to the PSE toolkit information system to discover the needed resources.

As a final remark, it must be specified that all the operations described in this section can be performed with the aid of a graphical interface which hides to the user the technical details regarding the ontology and XML schema formalisms. As an example, the DAMON ontology (Cannataro and Comito 2003), described in Section “Ontology System”, offers a graphical tool through which a user can browse the data mining ontology by simply selecting the graphical objects associated to ontology concepts and relationships. This tool gradually presents deeper levels of the ontology: the user starts at the top of the ontology and can navigate towards more specific topics by clicking the classes of interest (diving into the information). At any point, the map shows the current class, its parent and its subclasses.

ONTOLOGY SYSTEM

“An ontology is an explicit specification of a conceptualization” (Gruber, 1993); it is a shared understanding of some domains of interest, which is often conceived as a set of classes (concepts), relations, functions, axioms and instances. Concepts in an ontology are usually organized in taxonomies. Each taxonomy, for a given application domain, organizes the concepts and terms into a classification structure.

To produce satisfactory results, a PSE should be designed using the best domain practice and following decisions made by skilled engineers in practical situations. Ontologies can be used for managing the knowledge in a Grid-based PSE environment allowing for the building of semantically enriched knowledge bases.

In the proposed information system, a suite of ontologies are used to classify resources and components provided by the PSE toolkit and, as a PSE toolkit is tailored towards different application domains, different ontologies are used to manage specific knowledge in different domains.

Due to the large heterogeneity of resources, two types of classifications are needed:

- *domain-independent* classification: resources are classified into generic types of resources, e.g. data sources, software, hardware resources, applications, Web/Grid services, etc.
- *domain-dependent* classification: generic classes of resources are instantiated or specialized into domain specific classes of resources.

Accordingly, each ontology should be composed of two parts, a specific domain part and a core part. The core part models knowledge related to generic PSE components/resources which are common to different application domains and provides domain independent primitives to build domain specific ontology. The domain specific part is an explicit description of domain specific terms, characteristics, components, and relationships among them. Moreover the description of relevant tasks of a domain (e.g. retrieval and analysis) should also be modeled. To this aim all the different aspects of the domain should be modeled: domain specific knowledge, domain specific tasks and applications.

Ontologies are modeled as a set of taxonomies derived from the specialization of a number of basic classes. These taxonomies may be linked together via *relations* or *axioms*. Two kinds of *relations* are used to organise ontological knowledge in the domain:

- *specialisation relation* ("is-a"): specialises general concepts in more specific ones. An "is-a" relation states that a class A is a subclass of B if every instance of A is also an instance of B.
- *"has part" relation*: defines a partition as a subclass of a class.

As the necessity of building a PSE for a specific application domain emerges, an ontology of concepts related to this domain should be added to the PSE toolkit, so that the proposed information system can exploit such an ontology in order to describe and categorize the resources used in that domain. In particular, in the last few years we developed an ontology for the data mining domain and an ontology for the bioinformatics application domain. In the following, we briefly introduce the *DAMON* ontology (*DAta Mining ONtology*) (Cannataro and Comito 2003) and the *Bioinformatics* ontology (Cannataro, Comito et al 2004), and describe how such ontologies are employed in the PSE toolkit. Similarly, other ontologies for different application domains can be retrieved from the literature and integrated in the ontology system of the PSE toolkit.

DAMON is an ontology of the Data Mining domain. The main concepts modeled in DAMON are the following:

- A *Task* represents a data mining technique for extracting patterns from data. A task specifies the goal of a data mining process.
- A *Method* is a data mining methodology used to discover the knowledge. It can be thought as a structured manipulation of the input data to extract knowledge.
- An *Algorithm* is the programmatic procedure which performs a data mining task.
- A *Software* is an implementation of a data mining algorithm.
- A *Suite* implements a set of data mining algorithms: every algorithm may perform different tasks and employ different methods to achieve the goal.

Figure 3 shows the taxonomy obtained by creating subclasses of the data mining Task. Figures 4 to 6 show the taxonomies related to the other basic concepts (Method, Algorithm, and Software) of our ontological model; in each of these taxonomies we construct a subsequent specialization level for every task identified in the taxonomy of Figure 3.

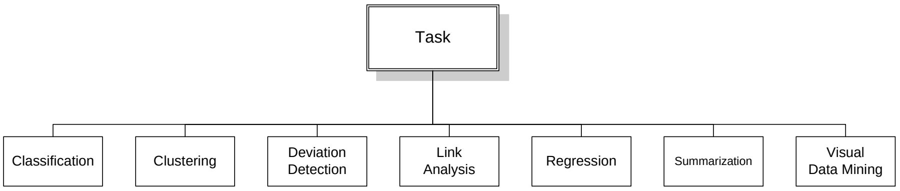

Figure 3. Data Mining Task taxonomy

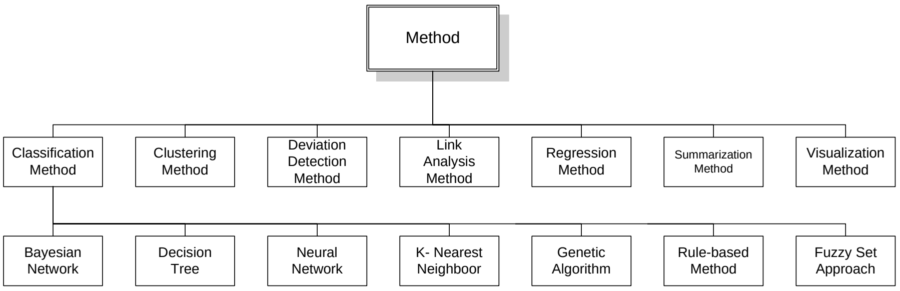

Figure 4. Data Mining Method taxonomy

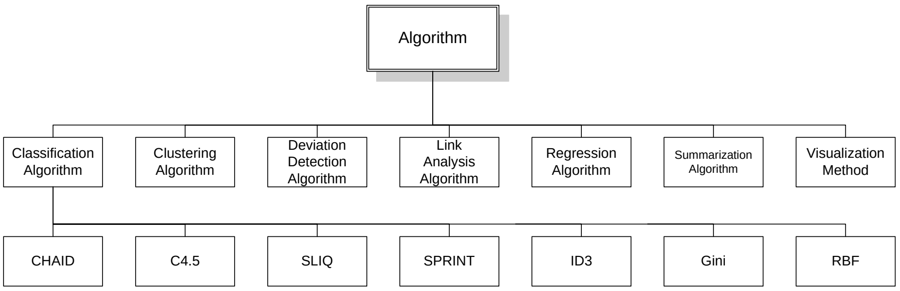

Figure 5. Data Mining Algorithm taxonomy

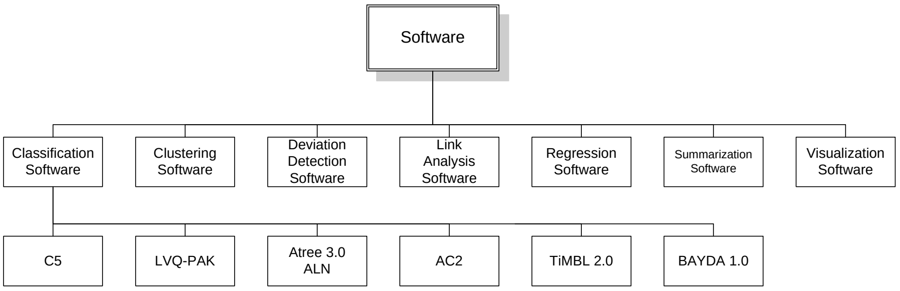

Figure 6. Data Mining Software taxonomy

Figure 7 shows an extract from the DAMON ontology regarding the conceptualization of the TribeMCL *Clustering Software*. The figure illustrates the hierarchical concept classifications (is-a relations) and the relationships (realized by means of *properties*) among concepts belonging to different taxonomies. In the example, TribeMCL *is a Clustering software* that implements the *Markov Clustering* algorithm. Such an algorithm is a *Clustering Algorithm* performing (*PerformsTask* property) the *Clustering* task and using (*UsesMethod* property) a *Statistical Analysis* method that is constrained to be a *Clustering Method* specifying (*SpecifiesTask* property) the *Clustering* task.

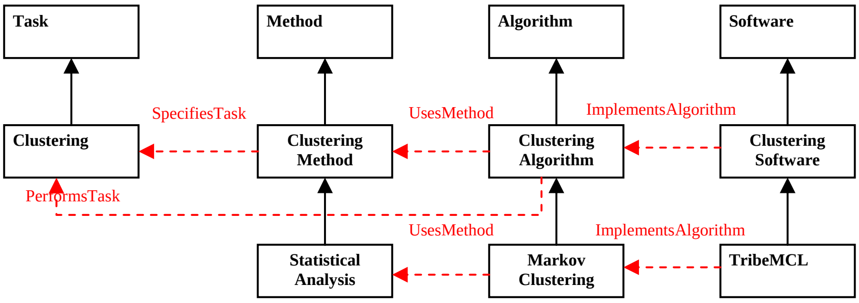

Figure 7. A fragment of the DAMON ontology for the TribeMCL software

The *Bioinformatics* ontology integrates different aspects of bioinformatics, including computational biology, molecular biology and computer science. In such an ontology we classify the following bioinformatics resources:
1) biological data sources, such as protein databases (e.g.,SwissProt, PDB);
2) bioinformatics software components, such as tools for retrieving and managing biological data (e.g., SRS, Entrez, BLAST, EMBOSS);
3) bioinformatics processes/tasks (e.g. sequence alignment, similarity search, etc.).

Biological data sources are classified on the basis of the following features:

- the kind of biological data (e.g., proteins, genes, DNA);
- the format in which the data is stored (e.g., sequence, BLAST proteins sequence);
- the type of data source (e.g., flat file, relational database, etc);

Bioinformatics processes and software components are organized on the basis of the following parameters:

- the *biological function* achieved by the software; that is the specific bioinformatics task (e.g., sequence analysis, secondary structure prediction, etc);
- the methodology (method) that the software uses to perform a bioinformatics task (e.g. GOR, Chou and Fasman, etc.)
- the algorithm implemented by the software (e.g. Clustalw, SmithWaterman, etc.);
- the data source on which the software works on (e.g. Swiss-Prot, PDB, etc.);
- the kind of output produced by the software;
- the software components used to perform a task (e.g. BLAST, EMBOSS, etc.).

As for the DAMON ontology, a taxonomy that specializes each of these classification parameters is implemented. For example the data source taxonomy classifies the different databases specifying the kind of biological data stored, the format in which the data is stored, the type of data source (flat file, relational database, and so on), etc.
Figure 8 shows a fragment of the bioinformatics ontology. The ontology can be explored by choosing one of the previous classifying parameters. For example, exploring the biological function taxonomy it is possible to determine for a given function which are the available algorithms that achieve it, and then which software implements the chosen algorithm. Moreover it is possible to find the data sources and the biological elements involved in that function.

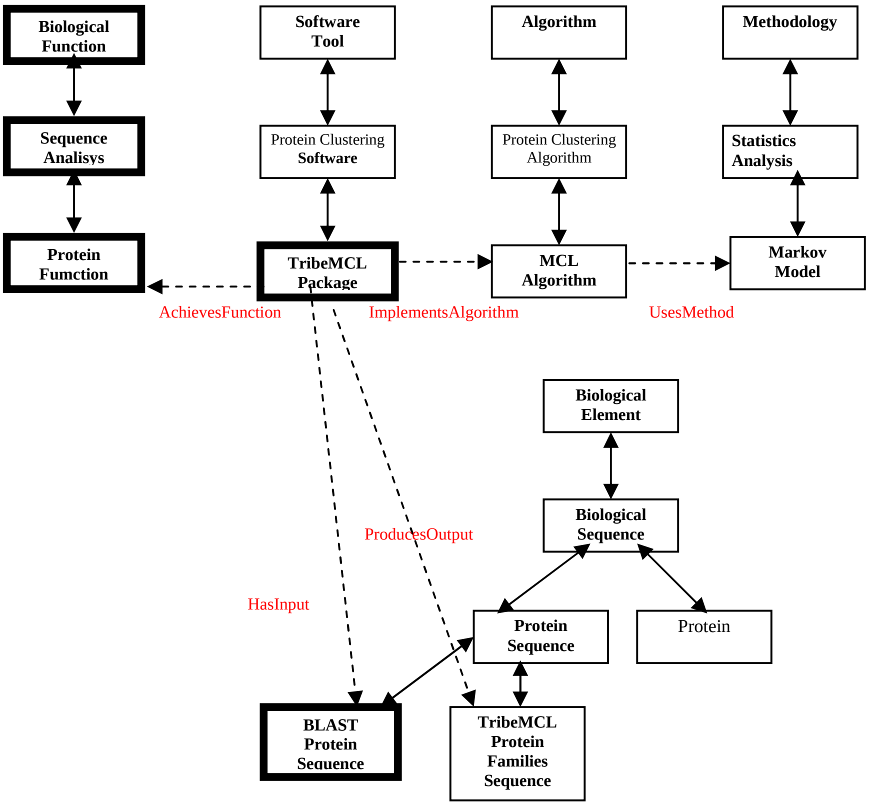

Figure 8. A fragment of the bioinformatics ontology for the TribeMCL software

As explained before, for each resource category, the ontology system generates the *structure* of the *semantic* and *resource* metadata sections.

Ontologies are maintained as OWL files in centralized/hierarchical repositories. To be stored in a WSRF Web service, the ontology must be written in an XML document conform to an XML schema. Though it is not possible to represent all the details of an ontology structure in the XML schema formalism, we used a translation mechanism that preserves as much information as possible in an XML schema and in the compliant XML documents.

In the case of semantic metadata, we will exploit the pertinent domain ontology to characterize the given resource type in that particular domain. For example, in order to define the semantic metadata of the category <*software, scientific data mining*>, we characterize the *software* resource type in the *data mining* domain by exploiting the DAMON ontology. Accordingly to the fragment shown in Figure 7, we explore the DAMON ontology following all the relationships related to the *software* concept. More precisely the “is-a” relations within a same taxonomy are encoded through parent-child XML relationships, whereas relations among different taxonomies are encoded by associating an XML element to each of them.

Figure 9 reports an extract from the XML schema `SciDataMiningTools.xsd`, which defines the structure of semantic metadata for a data mining software. The schema specifies that the root element of every conform XML document must contain the element `DataMiningSoftware` which specifies, in its sub-elements, the kind of task that is performed by the software and the kind of algorithm which is implemented. The possible values for such sub-elements are those shown in Figures 3 and 5. Furthermore the `ImplementsAlgorithm` element permits to specify the method used by the software, among those shown in Figure 4.

Note that the first part of the semantic metadata section shown in Figure 1 is validated against the schema SciDataMiningTools.xsd, since it is related to a data mining sofware.
In the same way, exploiting the Bioinformatics ontology shown in Figure 8, we obtain the XML schema BioinformaticsSoftware.xsd (see Figure 10) which defines the structure of semantic metadata for the category <*software, bioinformatics*>. The second part of the semantic metadata section shown in Figure 1 is validated against the schema BioinformaticsSoftware.xsd, since it is related to a bioinformatics sofware.

```
<schema targetNamespace="http://domain/path/SciDataMiningTools"
    xmlns="http://www.w3.org/2001/XMLSchema" …>
  <complexType name="DMSoftwareType">
    <sequence>
    <element name="PerformsTask" type="TaskType"/>
    <element name="ImplementsAlgorithm" type="AlgorithmType"/>
    </sequence>
    <attribute name="name" type="string"/>
  </complexType
  <complexType name="AlgorithmType">
    <element name="UsesMethod" type="MethodType"/>
    <attribute name="name" type="string"/>
    <attribute name="kind" type="AlgoCategory"/>
  </complexType>
  <complexType name="MethodType">
    <attribute name="name" type="string"/>
    <attribute name="kind" type="MethodCategory"/>
  </complexType>
  <simpleType name="TaskType">
    <restriction base="string"
      <enumeration value="Clustering"/>
      <enumeration value="Classification"/>
      <enumeration value="Association Rules"/>
      …
    </restriction>
  </simpleType>
  <simpleType name="AlgoCategory">
    <restriction base="string"
      <enumeration value="ClusteringAlg"/>
      <enumeration value="ClassificationAlg"/>
      <enumeration value="Association RulesAlg"/> …
    </restriction>
  </simpleType>
  <simpleType name="MethodCategory">
    <restriction base="string"
      <enumeration value="ClusteringMethod"/> …
    </restriction>
  </simpleType>
  <element name="SemanticMetadata"/>
   <complexType>
    <element name="DataMiningSoftware" type="DMSoftwareType"/>
   </complexType>
  </element>
</schema>
```

Figure 9. An extract from the XML schema SciDataMiningTools.xsd

```
<schema targetNamespace="http://domain/path/BioinformaticsSoftware"
    xmlns="http://www.w3.org/2001/XMLSchema" …>
  <complexType name="BioSoftwareType">
    <sequence>
    <element name="BiologicalFunction" type="FunctionType"/>
    <element name="BiologicalElement" type="ElementType"/>
    <element name="HasInput" type="string"/>
    <element name="ProducedOutput" type="string"/>
    </sequence>
    <attribute name="name" type="string"/>
  </complexType
  <complexType name="FunctionType">
    <attribute name="name" type="string"/>
    <attribute name="kind" type="FunctionValue"/>
  </complexType>
  <complexType name="ElementType">
    <attribute name="name" type="string"/>
    <attribute name="kind" type="ElementValue"/>
  </complexType>
  <simpleType name="FunctionValue">
    <restriction base="string"
      <enumeration value="SequenceAnalysis"/>
      <enumeration value="ProteinFunctionPrediction"/>
      …
    </restriction>
  </simpleType>
  <simpleType name="ElementValue">
    <restriction base="string"
      <enumeration value="Protein"/>
      <enumeration value="Gene"/>
      …
    </restriction>
  </simpleType>
  <element name="SemanticMetadata"/>
   <complexType>
    <element name="BioinformaticsSoftware" type="BioSoftwareType"/>
   </complexType>
  </element>
</schema>
```

Figure 10. An extract from the XML schema `BioinformaticsSoftware.xsd`

CONCLUSIONS

A Grid PSE toolkit based on Web services is a group of technologies that allows for building PSEs for different application domains by exploiting the features and functionalities of both the Web service paradigm and the Grid infrastructure. Such PSE toolkits require an efficient approach to manage the heterogeneity of the involved resources. The paper proposes a metadata model that allows for classifying and describing resources needed for different domains. A metadata document, associated to each resource, includes an ontological metadata section that identifies the resource category, a semantic metadata section that characterizes resources in different application domains and assists discovery services, and a resource metadata section that gives details about how to use and access a resource. Moreover, the paper described the architecture of an information system that allows for a uniform and flexible management of metadata. The information system exploits the basic information services of a Grid framework based on Web services (i.e. the WSRF framework) to aggregate and index metadata.

Currently the information system is usable in the bioinformatics and data mining application domains, since the related domain ontologies have already been integrated. We plan to integrate more domain ontologies, starting from the geo-computation domain. Moreover, we are going to evaluate the performance of the information system by issuing a large set of resource discovery requests in the bioinformatics and data mining domain. On the basis of performance results, we will focus our future work on the improvement and optimization of the PSE toolkit high-level services.

ACKNOWLEGMENTS

This work has been partially supported by the Italian MIUR FIRB Grid.it project RBNE01KNFP on High Performance Grid Platforms and Tools and by the FP6 Network of Excellence CoreGRID (Contract IST-2002-004265).

REFERENCES

Aktas, M. S., Pierce, M., Fox, G. F. (2004). Designing Ontologies and Distributed Resource Discovery Services for an Earthquake Simulation Grid, *Proceedings of the GGF11 Semantic Grid Applications Workshop*, Honolulu, USA (2004) 1-6.

Cannataro, M., Comito, C. (2003). A Data Mining Ontology for Grid Programming, *Proceedings 1st Int. Workshop on Semantics in Peer-to-Peer and Grid Computing (SemPGrid2003)*, Budapest, Hungary, 113-134.

Cannataro, M., Comito, C., Congiusta, A., Folino, G., Mastroianni, C., Pugliese, A., Spezzano, G., Talia, D., Veltri, P.(2004). A General Architecture for Grid-Based PSE Toolkits, Workshop on State-of-the-Art in Scientific Computing, *Proceedings of the International Workshop on State-of-the-Art in Scientific Computing (PARA'04)*, Copenhagen, Denmark.

Cannataro, M., Comito, C., Congiusta, A., Veltri, P. (2004). PROTEUS: a Bioinformatics Problem Solving Environment on Grids, *Parallel Processing Letters (PPL)*, 14(2), 217-237, World Scientific Publishing Company.

Curbera, F., Goland, Y., Klein, J., Leymann, F., Roller, D., Thatte, S., Weerawarana, S (2005). Business Process Execution Language for WS. Retrieved January 8, 2006, from http://www-128.ibm.com/developerworks/webservices/library/ws-bpel/index.html.

Deelman, E., Blythe, J., Gil, Y., Kesselman, C., Mehta, G., Vahi, K., Blackburn, K., Lazzarini, A., Arbree, A., Cavanaugh, R., Koranda, S. (2003). Mapping Abstract Complex Workflows onto Grid Environments, *Journal of Grid Computing*, 1(1), 25-39, Kluwer Academic Publishers, Netherlands.

Enright A.J., Van Dongen S., Ouzounis C.A.: TribeMCL (2004). An efficient algorithm for large scale detection of protein families. Retrieved November 21, 2005, from http://www.ebi.ac.uk/research/cgg/tribe/.

Foster, I, Kesselman, C., Nick, J., Tuecke, S. (2002). The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, Globus Project. Retrieved November 21, 2005, www.globus.org/research/papers/ogsa.pdf .

Fox, G. (2003). Data and Metadata on the Semantic Grid, *Computing in Science and Engineering*, 5(5).

Gallopoulos, E., Houstis, E. N., Rice, J. (1994). Computer as Thinker/Doer: Problem-Solving Environments for Computational Science, *IEEE Computational Science and Engineering*, 1(2).

GGF, the Global Grid Forum (2005). Retrieved January 14, 2006, from http://www.ggf.org.

Globus, the Globus Toolkit (2005). Retrieved January 14, 2006, from http://www.globus.org.

Gruber, T., R. (1993). A translation approach to portable ontologies, *Knowledge Acquisition*, 5(2), 199-220.

Hastings, S., Langella, S., Oster, S., Saltz., J. (2004). Distributed Data Management and Integration: The Mobius Project, *Proceedings of the GGF11 Semantic Grid Applications Workshop*, Honolulu, USA, 2004, 20-38.

Mastroianni, C., Talia, D., Trunfio, P. (2003). Managing Heterogeneous Resources in Data Mining Applications on Grids Using XML-Based Metadata, *Proceedings of the IPDPS 2003*, IEEE Computer Society Press, 2003.

The OGSA-DAI Project (2005). Open Grid Services Architecture Data Access and Integration. Retrieved June 2, 2005, from http://www.ogsadai.org.uk/.

OWL, the OWL Web Ontology Language Reference, W3C Recommendation (2004). Retrieved June 20, 2005, from http://www.w3.org/TR/owl-ref/.

OWL-S, the OWL Services Coalition (2005). Semantic Markup for Web Services. Retrieved June 20, 2005, from http://www.daml.org/services/owl-s/1.0/owl-s.html.

The Semantic Grid project (2005). Retrieved September 5, 2005, from http://www.semanticgrid.org.

WSRF, the Web Services Resource Framework (2005). Retrieved January 20, 2006, from http://www.globus.org/wsrf/.

Xindice, the Apache Xindice (2005). Retrieved October 12, 2005, from http://xml.apache.org/xindice.

ABOUT THE AUTHORS

Carmela Comito is pursuing her PhD in Systems and Computer Engineering at the University of Calabria, Italy. She received her Laurea degree in Computer Engineering from the same university. Her current research interests include Grid computing, Peer-to-Peer data Management, Metadata Management and Distributed Databases.

Carlo Mastroianni is a researcher at the Institute for High Performance Computing and Networks of the Italian National Research Council (ICAR-CNR) in Cosenza, Italy. From 1999 to 2001, we worked as a Computer Engineer at the Computer Department of the Prime Minister Office, in Rome. He received a PhD in Computer Engineering from the University of Calabria, Italy, in 1998. Currently, he teaches Computer Networks at the University of Calabria and Databases at the “Parthenope” University of Naples. Mastroianni published more than 40 papers in international journals and conference proceedings.

Domenico Talia is a full professor of computer science at the Faculty of Engineering at the University of Calabria, Italy, a research associate at ICAR-CNR in Rende, Italy and a partner at Exeura s.r.l. He received the Laurea degree in Physics at University of Calabria. His research interests include grid computing, distributed knowledge discovery, parallel data mining, parallel programming languages, cellular automata, and peer-to-peer systems.
Talia published three books and more than 150 papers in international journals such as *Communications of the ACM*, *IEEE Computer*, *IEEE TKDE, IEEE TSE, IEEE TSMC-B, IEEE Micro, ACM CS, FGCS, Parallel Computing, IEEE Internet Computing* and conference proceedings. He is a member of the editorial boards of the *IEEE Transactions on Knowledge and Data Engineering, Future Generation Computer Systems* journal, the *International Journal on Web and Grid Services*, the *Parallel and Distributed Practices* journal, and the *Web Intelligence and Agent Systems International* journal. He is a member of the Executive Committee of the CoreGRID Network of Excellence and member of the European Knowledge Discovery Network of Excellence. He is serving as a program committee member of several conferences and is a member of the ACM and the IEEE Computer Society.