

# Metadata Management in a Grid-based PSE Toolkit

Carmela Comito<sup>1</sup>, Carlo Mastroianni<sup>2</sup> and Domenico Talia<sup>1</sup>

<sup>1</sup>DEIS, University of Calabria, Via P. Bucci 41 c, 87036 Rende, Italy  
{ccomito, talia}@deis.unical.it

<sup>2</sup>ICAR-CNR, Via P. Bucci 41 c, 87036 Rende, Italy  
mastroianni@icar.cnr.it

**Abstract.** A PSE toolkit can be deemed as a group of technologies within a software architecture through which multiple PSEs can be built for different application domains. This paper presents a metadata model for Grid-based PSE toolkits and the architecture of an information system based on the metadata management model. These two components contributed to the definition of a general model of metadata management for supporting the design and implementation of PSEs on Grids.

## 1 Introduction

A Problem Solving Environments (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 [9]. 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 can be deemed as 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 and tools. One of the most interesting models in the area of parallel and distributed computing is the Grid.

Through the combination of PSE toolkit issues and the exploitation of Grid features and functionalities we obtain the possibility to design Grid-based PSE toolkits. The effective use of a Grid-based PSE requires the definition of an approach to manage the heterogeneity of the involved resources that can include computers, data, network facilities, sensors, and software tools provided by different organizations [3]. Heterogeneity arises mainly from the large variety of resources within each category. The management of such resources requires the use of metadata that, through an accurate categorization of resources, provides useful information about the features of resources and their effective use.

Metadata is used to classify and manage a resource, but classification parameters, i.e. the structure of metadata information, depend on the type of the resource (i.e. software, hardware, data etc.) and on the application domain in which it is used. An information system of a PSE toolkit should use a uniform approach to manage metadata documents having different structures. Furthermore, such a toolkit should exploit information services provided by the underlying Grid framework, e.g. by the Index Services of the Open Grid

---

This work was partially supported by the Italian MIUR FIRB Grid.it project RBNE01KNFP on High Performance Grid Platforms and Tools.

Services Architecture (OGSA) [6], or by the GroupServices of the Web Services Resource Framework (WSRF) [11].

We designed both a metadata model and an information system that can be used in a Grid-based PSE toolkit to offer a uniform and at the same time flexible approach to the management of metadata. The paper is organized as follows. Section 2 discusses related work. Section 3 describes the metadata model and section 4 presents an architecture for the information system based on the metadata model. Section 5 concludes the paper.

## 2 Related Work

The key role of metadata management for the effective designing of distributed data mining systems is widely recognized in the literature [14, 8, 16]. A recent workshop report released within the e-Science project [14] highlights the benefits of adopting standard representation of metadata, based on the XML language, to face the issues coming both from increasing data volume and the heterogeneous and distributed nature of scientific data.

The adoption of the service oriented model in novel Grid systems, based on the OGSA architecture or the WSRF framework, has an impact on the management of metadata and on the architecture of information services, since in such systems services and resources are exposed as Grid services (also called WS-Resources in WSRF). The information model of service-oriented Grid frameworks is essentially based on two features:

(i) Metadata about Grid service instances is stored into XML-encoded documents, called Service Data Elements (SDE) in OGSA and Resource Properties in WSRF. In both cases, such documents must conform to enriched XML schema documents.

(ii) Information is collected and indexed by means of hierarchical information services (Index Services in OGSA and GroupServices in WSRF) that subscribe to the information stored in Grid services, aggregate it and provide it to high level browsing and querying services.

In the Grid computing community there is an effort to define the so called Semantic Grid [20], whose approach is based on the systematic description of resources through metadata and ontologies. In [8] the role of metadata in the context of the Semantic Grid is discussed. Here 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. When exploiting the OGSA architecture, based on the Grid Service technology [7], it is essential to integrate metadata embedded in services (i.e. information stored in the XML-based Service Data Elements provided by Grid Services) and metadata external to Grid Services, which can be stored in distributed databases and repositories with very variable scope and completeness.

Metadata management models have been proposed to address the requirements of problem solving environments. Examples of significant PSEs that use XML-based metadata models for the representation of heterogeneous resources are WebFlow and the Common Portal Application [13].

Reference [1] 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 [19]. This approach permits to exploit the richness of semantic languages, but does not take full advantage of the information services provided by service-oriented Grid frameworks.

In [12] 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.

### 3 A Metadata Model for a Grid-based PSE Toolkit

In a Grid-based PSE toolkit, metadata must be used to manage the heterogeneity that derives from the large variety of resources available within each resource class [3]. 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 resource *category*. In this context we define a category as the set of resources of a given type (e.g., software, data source, hardware etc.) which can be used in a given application domain.

We can identify at least the following types of resources:

- 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 system).
- Software component resources, among which Web and Grid services are gaining a major role.
- Hosts and other hardware devices (computers, physical memory, cpu speed, number of nodes in a parallel computer) as well as network characteristics (connection capabilities, network load, network latency, available network bandwidth).
- Applications modeled as workflows.

A metadata document associated to a resource is composed of three sections:

*Ontological* metadata is used to identify, for each resource, its type and the application domains in which it can be used. Ontological metadata allows for individuating the *structure* of the remaining sections of the metadata document. Ontological metadata is generated and managed by an ontology system.

*Semantic* metadata is used to characterize resources within a given category. Such a categorization is operated using the metadata structure document specified by ontological metadata.

*Resource* metadata supplies specific information about a resource to facilitate its usage. Resource metadata must conform to the metadata structure determined by ontological metadata for each resource type. It is further classified into *description* and *usage* metadata.

The distinction of three metadata sections has been proposed for two main reasons:

- i. In a PSE toolkit, resources should be annotated at different levels and in different times. When publishing a resource, it is necessary to specify the category to which it belongs. Furthermore, a resource should be classified within its category to facilitate key services

- such as resource discovery and workflow composition. Further metadata information is provided to facilitate the use of a resource.
- ii. The differentiation of metadata information is useful to take advantage of the benefits offered both by the Grid technology and the by ontology systems and languages. Grid technology permits to store metadata information within a Grid service. This way, it is possible to exploit Grid information services to discover and access resources. However, the expressive power of such metadata is limited by the XML Schema formalism adopted by service-oriented Grid frameworks, like OGSA and WSRF. The solution proposed is as follows: metadata is stored in XML documents conform to XML schemas (hence, such documents can be stored within Grid services), whereas an ontology system uses more expressive ontological languages, such as OWL [19], to classify application domains and define the structure of such domains.
- A last type of metadata information is *state* metadata that stores the state of a dynamic resource. However, state metadata cannot be stored in a metadata document separated by the resource, but it must be encapsulated within the resource.

### 3.1 Ontological metadata

The ontological metadata section specifies the categories to which a resource belongs (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 example, `TribeMCL` [5] is a software used in the bioinformatics domain to perform data mining computations. Ontological metadata should specify the type of that resource (i.e., service-oriented software) and the involved application domains (bioinformatics and data mining). The ontological section of the metadata document related to `TribeMCL` is:

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

The element `<ResourceType>` specifies that the resource is a software, and that such software 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 determine 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: `DataMiningSoftware.xsd`, and `BioinformaticsSoftware.xsd`.

### 3.2 Semantic metadata

Semantic metadata includes information that characterizes the resources within a given category, and that can be used to facilitate the discovery and browsing of resources.

Meaningful properties are defined by the system ontology for each category of resources, and are used to specify resource characteristics, functionalities and purpose. Accordingly, a set of parameters and possible associated values are specified by means of an XML schema generated by the ontology system.

For example, if the category of data mining software is defined, the ontology system permits to determine the parameters and values that can be used to characterize an instance of that class. The ontology system produces the XML schema `DataMiningSoftware.xsd`, to which semantic metadata must conform. Furthermore, the ontology system specifies the values of ontological metadata parameters that will be used to individuate the resource category, as shown in Section 3.1.

An extract from the XML schema `DataMiningSoftware.xsd` is reported in Figure 1. The schema defines five elements that are used to categorize a data mining software and specifies, through the definition of the corresponding XML schema types, the values that can be assigned to those elements. In particular, semantic metadata elements permit to specify the kind of input data sources, the sort of knowledge that can be discovered, the type of technique adopted, the algorithm implemented and the driving method exploited by the mining process.

```
<schema targetNamespace="http://domain/path/DataMiningSoftware"
  xmlns="http://www.w3.org/2001/XMLSchema" ...>
<simpleType name="KindOfKnowledge_value">
  <restriction base="string">
    <enumeration value="association rules"/>
    <enumeration value="clusters"/>
    <enumeration value="characteristics rules"/>
    <enumeration value="classification rules"/>
    ...
  </restriction>
  ...
</simpleType>
...
<element name="SemanticMetadata">
  <complexType>
    <sequence>
      <element name="KindOfKnowledge" type="KindOfKnowledge_value" minOccurs="0"/>
      <element name="KindOfData" type="KindOfData_value" minOccurs="0"/>
      <element name="KindOfTechnique" type="KindOfTechnique_value" minOccurs="0"/>
      <element name="Algorithm" type="Algorithm_value" minOccurs="0"/>
      <element name="DrivingMethod" type="DrivingMethod_value" minOccurs="0"/>
    </sequence>
  </complexType>
</element>
</schema>
```

**Figure 1.** An extract from the XML schema `DataMiningSoftware.xsd`

An extract from the XML schema `BioinformaticSoftware.xsd` is reported in Figure 2. This schema describes the structure of semantic metadata for a software used in the bioinformatics domain. It defines four elements that specify the biological function achieved by the software, the biological element analyzed, the kind of biological data received as input and the kind of biological data produced as output.

If a resource belongs to more than one resource category (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 of the software `TribemCL`, discussed in Section 3.1, is validated against the XML schemas `DataMiningSoftware.xsd` and `BioinformaticsSoftware.xsd`. Semantic metadata, reported in Figure 3, specifies that the software analyzes BLAST protein sequences extracted from a relational database in order to predict the protein function, uses a statistical method by implementing the Markov

Clustering algorithm (MCL), produces clusters in the form of TribeMCL protein families, and is executed through an automatic process.

```
<schema targetNamespace="http://domain/path/BioinformaticsSoftware"
  xmlns="http://www.w3.org/2001/XMLSchema" ...>
<simpleType name="BioFunction_value">
  <restriction base="string">
    <enumeration value="sequence analysis"/>
    <enumeration value="protein function prediction"/>
    ...
  </restriction>
</simpleType>
<simpleType name="BioElement_value">
  <restriction base="string">
    <enumeration value="protein"/>
    <enumeration value="gene"/>
    ...
  </restriction>
</simpleType>
...
<element name="SemanticMetadata">
  <complexType>
    <sequence>
      <element name="BiologicalFunction" type="BioFunction_value" minOccurs="0"/>
      <element name="BiologicalElement" type="BioElement_value" minOccurs="0"/>
      <element name="HasInput" type="KindOfInput_value" minOccurs="0"/>
      <element name="ProducedOutput" type="KindOfOutput_value" minOccurs="0"/>
    </sequence>
  </complexType>
</element>
</schema>
```

**Figure 2.** An extract from the XML schema BioinformaticsSoftware.xsd

```
<SemanticMetadata xmlns="http://domain/path/DataMiningSoftware" ...>
  <KindOfData>relational database</KindOfData>
  <KindOfKnowledge>clusters</KindOfKnowledge>
  <KindOfTechnique>statistics</KindOfTechnique>
  <Algorithm>MCL algorithm</Algorithm>
  <DrivingMethod>autonomous knowledge miner</DrivingMethod>
</SemanticMetadata>
<SemanticMetadata xmlns="http://domain/path/BioinformaticsSoftware" ...>
  <BiologicalFunction>protein function prediction</BiologicalFunction>
  <BiologicalElement>protein</BiologicalElement>
  <HasInput>BLAST protein sequence</HasInput>
  <ProducedOutput>TribeMCL protein families</ProducedOutput>
</SemanticMetadata>
```

**Figure 3.** Semantic metadata section of the software TribeMCL

### 3.3 Resource metadata

Resource metadata describes the main features and the modalities for accessing and using a resource, and can also be used to evaluate the quality of a resource. The structure of resource metadata does not depend on the particular application domain. Conversely, such a structure is determined by the resource type and defined with an XML schema associated to that resource type. As an example, software offered as a service can be annotated with metadata information such as: the URL of the service, the syntactic description of inputs and outputs etc.

Resource metadata is divided into *Description* and *Usage* metadata.

*Description metadata* provides a concise description of the service. It contains provider and contact information that refers to the entity that provides the resource and is responsible

for running the associated services. Furthermore, description metadata can include a functional description of the resource expressed in terms of the capabilities and functionalities offered by the resource. Information about the quality rating of the resource can also be provided. Finally, description metadata can provide information about the past usage of the 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 how to access and use a resource. Even if it would be preferable that all or most of resources were offered as services (Web or Grid 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, two different XML schemas are used. 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 that can be defined by command line arguments or by an Application Program Interface.

In the following, for three important types of resource (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 these types of resources, and when those standards are not sufficient, propose further formalisms: see reference [15] for more details.

### 3.3.1 Software components

Resource metadata is 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 separately discussed.

*Service-oriented software.* The *usage* metadata section must contain at least a reference to the WSDL document describing the service. However, the WSDL language cannot give semantic information about Web services due to the limited expression power of the XML Schema formalism. Currently, there are several proposals on how to semantically describe a software that is provided as a service. One important proposal has been formulated by the DARPA Agent Markup Language (DAML) Program [18]. The Semantic Web Services arm of the DAML program has 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:

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

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

*Non service-oriented software.* Resource metadata should contain the same type of information that is provided by OWL-S and WSDL documents for service-oriented resources: structure of input and output, information about the software provider, functional

description etc. Such information is encoded in an XML document. Details on the syntactic description of a software interface are given in [15].

### 3.3.2 Data resources

Data-related resources can be classified as follows:

1. *Data resource managers* are systems designed to manage data. Example are a file management 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.

*Non service 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 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) [17] 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.

### 3.3.3 Workflows

A main purpose of a Grid-based PSE toolkit is to facilitate users in the specification of complex applications that compose multiple tasks arranged in a workflow.

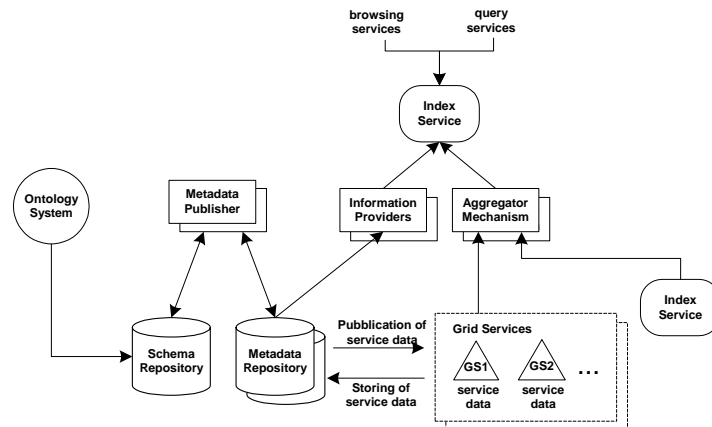
In our system, workflows can be defined at two levels: a *concrete workflow* contains only well defined resources (e.g. particular software resources to be executed on specified hosts) whereas an *abstract workflow* contains at least an *abstract resource*, that is a resource defined by means of constraints on metadata properties (e.g., a software that extracts clusters from bioinformatics data). An abstract workflow must be instantiated to a concrete workflow before being executed on the Grid; instantiation 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 an XML formalism and about workflow instantiation are given in [15].

If an application is composed of Web or Grid services, a concrete workflow can be also expressed by one of the languages that are emerging for this purpose, in particular OWL-S [18] and BPEL [4]. An OWL-S *composite process* is one that maintains some state; each message the client sends advances the state through the process. BPEL defines a model and a grammar for describing the behaviour of a business process based on interactions between the process and its partners. Note that both OWL-S and BPEL define *abstract* and *concrete* workflows. However, such definitions are related to the possible specification of network bindings; if an abstract workflow is defined as one that contain abstract resources, OWL-S and BPEL workflows should be considered as concrete workflows.

#### 4. Architecture of the PSE toolkit Information System

The architecture of the information system designed for our Grid-based PSE toolkit is depicted in Figure 4. The information system accomplishes two main tasks: it manages metadata describing the resources provided by the toolkit, and supports high-level discovery services.



**Figure 4.** Architecture of the PSE toolkit information system

The information system is integrated with the Globus Toolkit [10], in order to take advantage of the services offered by that technology (browsing and indexing services, information providers etc.). At present, the system is integrated with the Globus Toolkit 3 (GT3) based on OGSA; minor modifications will be required to port the system to the WSRF-based Globus Toolkit 4 that is going to be released. In particular, in the Globus Toolkit 4 Grid services will be replaced by WS-Resources, service data by resource properties and Index Services by ServiceGroups.

The information system is composed both by 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 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 4, components that are inherently distributed are duplicated. In the rest of the section, details are given about the structure and functionalities of the information system components.

#### 4.1 Ontology System

Ontologies provide at the very least a taxonomy that organizes the concepts or terms into a classification structure. In the proposed information system, an ontology system is used to classify the resources and components provided by the PSE toolkit and individuate the structure of associated metadata. Due to the large heterogeneity of resources, two orders of classifications are necessary:

- *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: it performs a classification of application domains, guided by domain experts. The classification can individuate multiple and possibly overlapping sub-domains within a larger domain.

As mentioned in Section 3, the ontology system produces the *ontological* metadata that allows for individuating the categories to which a resource belongs by specifying the type of the resource and the application domains in which it is used. For each category, the ontology system generates the *structure* of the metadata information that will be associated to the resources belonging to that category. In particular, XML schemas are produced for each type of resource individuated by the domain-independent classification: *resource* metadata is validated against such schemas. Furthermore, an XML schema is generated for each couple <type of resource, application domain>: *semantic* metadata is validated against such a schema.

The ontology is maintained as an OWL file in a centralized/hierarchical repository.

#### 4.2 Schema Repository

The schema repository stores the XML schemas generated by the ontology system. It is used for two main purposes:

1. Editing of metadata documents. The schema repository is accessed in the semi-automatically metadata document editing process. In particular, when a resource belonging to a given resource category is created/modified, an XML schema is retrieved from the repository to individuate the parameters and values through which it is possible to characterize that resource.
2. Support to querying and browsing. The schema repository is accessed to assist querying and browsing over PSE components and services. If a user needs to discover resources having specified characteristics, she can directly use the Grid information service if she knows the XML schema that defines the structure of the semantic metadata section. Otherwise she can retrieve the XML schema from the repository and use the Grid information service to discover the needed resources.

### 4.3 Metadata Publisher

A metadata publisher is used to create/modify the metadata documents related to new/existing PSE resources. This component allows a user to view the characteristics of the resource categories defined by the ontology system, and the corresponding XML schemas stored in the schema repository.

If a user wants to publish a new resource, she verifies if that resource belongs to one of the resource categories described in the schema repository; if this is the case, the user selects the corresponding XML schema and edits the metadata document with an assisted procedure that guarantees the consistency of the document. If the new resource does not belong to any registered resource category, the user should use the ontology system to refine the classification of application domains, and possibly create a new resource category and a corresponding XML schema that will be stored in the schema repository. Afterwards the user will be able to use the new schema and produce the metadata document associated to the new resource.

### 4.4 Metadata Repository and Grid Services

The metadata repository stores the metadata documents related to the components/services provided by the PSE toolkit. As mentioned in Section 3, the choice of using XML schemas to define the structure of metadata allows for an efficient integration with the GT3 framework. Indeed the structure of the Service Data Elements (SDEs) stored within a Grid service is defined by means of enriched XML schemas (the Service Data Descriptions) associated to the WSDL document describing the service. As a consequence, the metadata document related to a service-oriented resource, or part of such a document, can be retrieved from the metadata repository and stored into a Grid service.

The opportunity 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 into 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 Grid service, metadata is retrieved by an *information provider* and published as SDEs. One information provider is associated to each Grid service, and is executed when the service is published for the first time and whenever metadata stored in the repository is modified by authorized users.

It is also possible that SDEs are modified during the lifetime of a Grid service. To avoid inconsistency problems, an attempt to modify metadata stored within an SDE requires an access to the corresponding document stored in the metadata repository. If the access is authorized, a lock is requested on the database, the requested modification is performed on the database with a synchronous operation and finally the SDE is modified as requested. Though this procedure could be time consuming, the frequency with which it is required is low, since metadata describing a resource is usually static. Notice that state metadata,

which instead can be very dynamic, is not part of the metadata document, therefore it is not stored into the metadata repository.

The metadata repository adopted in the PSE toolkit is a distributed XML database based on the Apache Xindice [2] 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 GT3 Index Services, as described in the next subsection.

#### 4.5 Index Services

The GT3 information system produces, aggregates and indexes metadata related to the resources provided by a set of Grid hosts belonging to a VO. Such a system exploits the functionalities of the OGSA Index Services [6]; usually each VO provides one Index Service, but more Index Services, organized in a hierarchy, can be installed on a large VO.

Index Services are used for browsing and querying metadata documents made available by the PSE toolkit. Metadata 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 retrieves the XML metadata documents stored in the metadata repositories of a VO, and publishes them into the Index Service of that VO.
2. *Service-oriented resources.* The Index Service subscribes to the SDEs that have to be aggregated and indexed, in order to be notified of changes. The GT3 *aggregator mechanism* is used to retrieve such data from the Grid services and publish it in the Index Service. If the Index Services of a VO are organized in two or more levels, the *aggregators* can retrieve data from lower level Index Services, and publish it in higher level Index Services, as depicted in Figure 4.

Since Index Services are fed with data retrieved both from Grid 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 those that are not service-oriented. Browsing and querying can be performed by means of the Globus Toolkit services (e.g. the Service Data Browser) or high level services provided by domain specific PSEs.

## 5 Conclusions

A Grid-based PSE toolkit is a group of technologies that allows for building PSEs for different application domains by exploiting the features and functionalities of a Grid infrastructure. A Grid-based PSE toolkit requires an efficient approach to manage the heterogeneity of the involved resources. The paper proposed a metadata model that permits to classify and describe resources needed for different domains. A metadata document, associated to each resource, includes an ontological metadata section that identifies the category of the resource, a semantic metadata section that characterizes the resource and is used to assist discovery services, and a resource metadata section that gives details about the access mechanisms. Moreover, the paper described the architecture of an information system that allows for a uniform and flexible management of metadata. The information system exploits an ontology system to semantically describe application domains and

resources, and the basic information services of a service-oriented Grid framework, namely the Globus Toolkit 3, to aggregate and index metadata.

## References

- [1] Aktas, M. S., Pierce, M., Fox, G. F.: Designing Ontologies and Distributed Resource Discovery Services for an Earthquake Simulation Grid, Proc. of the GGF11 Semantic Grid Applications Workshop, Honolulu, USA (2004) 1-6
- [2] The Apache XML: Apache Xindice, <http://xml.apache.org/xindice>
- [3] Cannataro, M., Comito, C., Congiusta, A., Folino, G., Mastroianni, C., Pugliese, A., Spezzano, G., Talia, D., Veltri, P.: A General Architecture for Grid-Based PSE Toolkits, Workshop on State-of-the-Art in Scientific Computing PARA 04, Copenhagen, Denmark (2004)
- [4] Curbera, F., Golland, Y., Klein, J., Leymann, F., Roller, D., Thatte, S., Weerawarana, S.: Business Process Execution Language for WS, <http://www-128.ibm.com/developerworks/webservices/library/ws-bpel/index.html>
- [5] Enright A.J., Van Dongen S., Ouzounis C.A.: TribeMCL: An efficient algorithm for large scale detection of protein families, <http://www.ebi.ac.uk/research/cgg/tribe/>
- [6] Foster, I., Kesselman, C., Nick, J., Tuecke, S.: Grid services for distributed system integration, *IEEE Computer*, 35(6) (2002) 37-46
- [7] Foster, I, Kesselman, C., Nick, J., Tuecke, S.: The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, Globus Project, [www.globus.org/research/papers/ogsa.pdf](http://www.globus.org/research/papers/ogsa.pdf) (2002)
- [8] Fox, G., Data and Metadata on the Semantic Grid, *Computing in Science and Engineering*, Volume 5, Issue 5, (2003)
- [9] Gallopoulos, E., Houstis, E. N., Rice, J.: Computer as Thinker/Doer: Problem-Solving Environments for Computational Science, *IEEE Computational Science and Engineering*, vol.1, n. 2 (1994)
- [10] The Globus Alliance: The Globus Toolkit, <http://www.globus.org>
- [11] The Globus Alliance: The Web Services Resource Framework (WSRF), <http://www.globus.org/wsrp/>
- [12] Hastings, S., Langella, S., Oster, S., Saltz, J.: Distributed Data Management and Integration: The Mobius Project, Proc. of the GGF11 Semantic Grid Applications Workshop, Honolulu, USA (2004) 20-38
- [13] Houstis, E., Catlin, A., Dhanjani, N., Rice, J., Dongarra, J., Casanova, H., Arnold, D., Fox, G., Problem-Solving environments, *The Parallel Computing Sourcebook*, M. Kaufmann Publishers (2002)
- [14] Mann, B., Williams, R., Atkinson, M., Brodli, K., Storkey, A., Williams, C.: Scientific Data Mining, Integration, and Visualization, report of the workshop held at the e-Science Institute, Edinburgh, <http://www.cacr.caltech.edu/~roy/papers/sdmiv-ltr.pdf> (2002)
- [15] Mastroianni, C., Talia, D., Trunfio, P.: Managing Heterogeneous Resources in Data Mining Applications on Grids Using XML-Based Metadata, *Proceedings IPDPS 2003*, IEEE Computer Society Press (2003)
- [16] The MyGrid project. <http://mygrid.man.ac.uk/myGrid/>
- [17] The OGSA-DAI Project: Open Grid Services Architecture Data Access and Integration, <http://www.ogsadai.org.uk/>
- [18] The OWL Services Coalition: OWL-S: Semantic Markup for Web Services, <http://www.daml.org/services/owl-s/1.0/owl-s.html>
- [19] The OWL Web Ontology Language Reference, W3C Recommendation 10 February 2004, <http://www.w3.org/TR/owl-ref/>
- [20] The Semantic Grid project: <http://www.semanticgrid.org>