Knowledge Perspectives in Data Grids

Luis Eliécer Cadenas, Emilio Hernández

Universidad Simón Bolívar, Departamento de Computación y T. I.,
Apartado 89000, Caracas 1080-A, Venezuela

Abstract. In this paper a methodology for accesing scientific data repositories on data grids is proposed. This methodology is based on ontology specification and knowledge representation. The concept of *Knowledge Perspective* is introduced, as the action of applying particular scientific conjectures or theories to the interpretation of experimental data and information. Data grid environments provide high levels of security and virtualization, which allow the users to create new data services on the data server side. These new services are based on the user's knowledge perspective. An implementation of this concept is presented, on a Globus-enabled Java execution platform.

1 Introduction

Computationally intensive technologies are very important in many areas of scientific research. These technologies are currently used to process, either locally or in distributed environments, considerable amounts of data and information. A new term has been coined to reference scientific research strongly dependent on computational and net-based collaboration: e-science [1]. Distributed platforms for data processing, increasingly known as grids, provide basic technologies for integrating multi-institutional sets of computational resources to support data processing. However, available tools are far from offering the levels of flexibility and capability required to transit the long way between data processing and knowledge generation. In this paper we propose and evaluate the concept of *knowledge perspective*, a tool for managing scientific data and experimental information in Data Grids environments. We define a knowledge perspective, or simply a *perspective*, as the consequence of applying a formalization of a theory to scientific data in order to help in the interpretation of experimental data and information.

In principle, scientific theories can be formalized as sets of universal quantified sentences, using First Order Logic (FOL). By selecting a set of such sentences we can define a theoretical framework (i.e an interpretation or viewpoint) for a specific experimental dataset. This selection may define relevant facts for the contrastation process of a particular theory. We can define, using FOL, concepts, properties, relations and sentences (i.e. closed formulas) that represent subsets of a particular scientific theory. In the context of processing a data source (or a combination of several data sources) for knowledge generation, there could be a first processing level in which the "raw" data is processed in

order to generate annotations and/or indexes. These indexes and annotations could highlight the relevant facts of the data according to the theory. In further processing levels the annotations can be semantically correlated in order to corroborate theories or conjectures.

The main contribution of this work is a computational model that allows the users to process data, in the context of Data Grids, which is epistemologically consistent with the nature of the scientific research activity. The users can safely create their own knowledge perpectives on the server or grid side, without the intervention of grid or system administrators. The operational base helps us manipulate and process efficiently very big distributed data sources in Data Grids. We implemented this model using SUMA/G [2], a distributed architecture for execution of Java programs which is implemented on top of Globus.

The rest of this paper is organized as follows. Section 2 formalizes the knowledge perspective concept and its relationship with the scientific and research activity. Section 3 introduces a general architecture to implement a knowledge perspective service in grids environments. Section 4 shows a practical example of the usage of this system to a bibliographic data source. Section 5 revises related work and section 6 offers our conclusions and future work.

2 Knowledge Perspectives

We define the concept of Knowledge Perspective from the definition of three sets. Lets Γ be the set that represents the objects x_i in the data source:

$$\Gamma = \{x_1, x_2, x_3, \ldots x_{n_2}\}$$

Given a set of predicates $P = \{p_1, p_2, p_3, \ldots p_{n_1}\}$, where each p_i represent attributes or relationships among elements of Γ, we can define Ω, which is a set of sets Φ_i:

$$\Omega = \{\Phi_1, \Phi_2, \Phi_3, \ldots \Phi_{n_1}\}$$

where the elements in each set Φ_i are tuples with elements in Γ satisfying the predicate p_i. Each p_i stands for a property or relationship in the ontology used to process the data source and could be organized in a taxonomical hierarchy. This hierarchy is described using description logic formalisms. This process is a first step to produce the knowledge perspective. Normally, elements of Ω (i.e sets Φ_i) are the product of annotating the data source using the concepts or properties p_i.

Λ is a possibly empty set of closed formulas (i.e. sentences) of predicate logic $A_i = W_i(x_1, x_2 \ldots x_{h_i})$. Each A_i represents conjectures or definitions about objects, properties and relationships in the data source, based on atoms p_i in P.

A Knowledge Perspective is then defined as an ordered tuple of sets:

$$\Pi = (\Gamma, \Omega, \Lambda)$$

In order to process a Knowledge Perspective we define at least two steps.

First, the annotation process over the data source, which consists in checking which objects are related through the predicate p_i. In order to do so, the user should provide the methods to verify each predicate over the data source. These methods are used to annotate the data source, probably producing indexes to objects having the property or standing in the relationship represented by p_i.

We define then the second stage of a knowledge perspective computation as the process of producing a set $\Omega^{'}$ using Ω and Λ. We can say that tuples $(\Gamma, \Omega, \Lambda)$ and $(\Gamma, \Omega^{'}, \Lambda)$ represent the same Knowledge Perspective. However, the validation of the conjectures A_i can be considered as the production of new knowledge, restricted to the data sources analyzed and using the vocabulary contained in P.

As an example to illustrate the previous definitions we can think of Γ as a data repository with astronomical images, Ω as a collection of sets of stars where each set has all the stars with the same apparent magnitude. Λ could be a set of predicate logic formulas (i.e. sentences or assertions in the theory) explaining the formation of supernovas, as a consequence of changes in the apparent magnitude within particular time frames. The computation of the apparent magnitude (i.e. the process to produce Ω) is done through an ontological annotation of the elements in Γ, and could be the product of processing the images or the result of using some existing catalogue.

3 Knowledge Perspectives implementation in Data Grids

We implement perspectives as new services, installed directly on the data source by the users. This is possible in data grids because of the security levels they provide. This approach has several advantages. Firstly, the user could send a short specification in a high level language (i.e. FOL) and the process is done at the data source. In this way it is possible to reduce the cost of data transfers. Secondly, it would facilitate data processing in places with legal restrictions for data transfers. In third place, it permits multiple views about the same data set. In this way different researchers or members of a comunity can share different points of view for the same data. Finally, new data services can evolve with the data source through updating mechanisms of the defined knowledge perspectives. Any data provider should offer, in addition to a normal data access service, a mechanism to process data *in-situ* and hosting services associated with data models installed by authorized users.

3.1 Architecture

The proposed architecture provides services to install new data queries and access services. These new services are built by processing the original data sets, providing in this way an additional perspective.

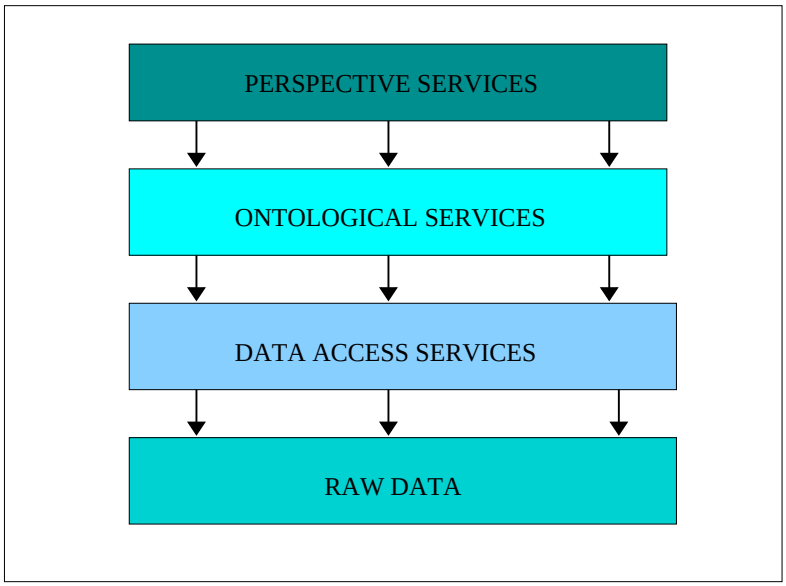

Fig. 1. Service Levels

Figure 1 shows the proposed architecture from the point of view of the services required in the Grid to offer perspective services. We defined the interfaces (API) required at each level and the related operational semantic. Through these interfaces we can virtualize the knowledge perspective service and integrate the same concept across many architectures, facilitating the deployment of distributed perspectives.

We propose a three layer architecture:

- The *Data Access Services* layer defines basic interfaces and the required services to access data sources. This service level would be typically installed by the data provider and offers abstractions to manipulate data sources, regardless the data format.
- The *Ontological Services* layer defines the interfaces and services required to create, store, manipulate and reason over ontologies. (i.e. UploadOntology, CheckOntology, etc). Using services at this level, the users can design an ontology which is adequate to their perspectives, with the required description of objects, properties and relationships. The users must then develop methods to produce the first level annotations. After this process the users obtain what we call the (*perspective 0*) level. Finally, the users develop the set of assertions (conjectures set and logical inferences) to be applied to *perspective 0* data in order to produce the *perspective 1* output.
- Finally, the *Perspective Manipulation Service* layer defines basic interfaces required to create, store and manipulate perspectives as objects, at both *perspective 0* and *perspective 1* levels (i.e. MakePerspective, QueryPerspective).

The execution of these services materialize both perspective levels by creating indexes using the data source and the user-provided ontology and theory. The first processing level establishes a match among objects in the data source and the satisfied predicates. The second one uses logical assertions in the Λ set to produce new satisfied inferences.

Currently, perspective, data and ontology services are defined and installed in SUMA/G[3], a grid infrastructure to execute Java bytecode in distributed environments, based on Globus services. In this software architecture we implemented a metaservice to install new services (SIMG). A *service* in this context is defined using a *service name*, a list containing all the services required to execute the new service (*requirements*), an *API*, a *documentation* and the set of packages that implements the service. The infrastructure offers great flexibility to install new services represented by java objects. This java object could have a constructor to annotate the data source and offers methods to access the annotated data source. For a future version we are defining a specialized *proxy* to query distributed databases processed using the perspective service. We provide facilities to query the data source, through the perspective service, using an option called *submit*. This option executes the queries asynchronously, and the results are stored temporarily in the execution agent. The user can ask at any time for these results using a mediator.

The service for installing new services directly by the users (SIMG) is crucial for the developing and installation of knowledge perspectives as defined in this work. The main reason is flexibility, because the users can process remote data transparently, i.e. in the same way they would process local data, in a secure way.

4 An example using Wordnet

As a proof of concept we implemented an example that allows us to improve data recovery from a Mysql database that contains information about scientific papers. We used Wordnet, a lexicographic reference system, available online [4]. The database was installed in an execution agent of SUMA/G, together with database access services, ontologies and perspectives as described in section 3.1. We used a Prolog version of Wordnet and developed a metainterpreter. The metainterpreter receives as input an english word and produces recursively as output an RDF file representing a taxonomical subtree with all the hiponyms of the word. This ontology is computed automatically by the metainterpreter and can be manipulated using primitives and methods provided through perspective and ontologies services. Using our perspective service we produce an index that points to papers which mention in the title any of the words contained in the hyponim tree.

In this example the data source is a relational database that contains information about scientific papers. Each table in this database represents a type of

object in the universe. We use an ontology (*science* [5]) to describe the objects represented in the data source. We defined an RDF Schema to describe in a generic way any relational database. This schema is shown in figure 2.

```
<rdf_:Tabla rdf:about="&rdf_;kb_db_00055"
 rdf_:Nombre_de_tabla="Profesores"
 rdf_:Representa="Science:Academic−Staff"
 rdf_:tamano="7912"
 rdfs:label="kb_db_00055">
<rdf_:tiene_atributos rdf:resource="&rdf_;kb_db_00056"/>
<rdf_:tiene_atributos rdf:resource="&rdf_;kb_db_00058"/>
<rdf_:tiene_atributos rdf:resource="&rdf_;kb_db_00059"/>
<rdf_:tiene_atributos rdf:resource="&rdf_;kb_db_00060"/>
</rdf_:Tabla>
<rdf_:Atributo rdf:about="&rdf_;kb_db_00056"
 rdf_:Longitud_atributo="50"
 rdf_:Nombre_Atributo="Science:First−Name"
```

Fig. 2. RDF description of the database

Through this schema we describe the objects in our data source and the meanings they stand for, using an ontology as reference (*science*). For example, the relation *Talkabout* could be defined in such a way that express the user's perspective. *Talkabout(paper,biology)* would mean that *paper* is a scientific paper about biology. In the predicate *Talkabout* the second argument is taken from a controlled vocabulary (i.e. the subtree of hyponim relationships produced through the metainterpreter). We want to process the data source to identify all the objects in the relationship *Talkabout*.

In this example, when we process a perspective, an index over the data source is generated. This index is an interpretation in the framework of a particular theory. Each sentence in the theory used to produce the perspective (each sentente A_i in Λ) generates a table with as many columns as the arity of A_i plus one column identifying the predicate. In our example, the only relation is hyponymy. The following sentences show a part of the subtree produced by the word *biology*:

$$\forall(X)Embriology(X) \rightarrow Biology(X)$$
$$\forall(X)Botany(X) \rightarrow Biology(X)$$
$$\forall(X)Phytology(X) \rightarrow Biology(X)$$

For this example our perspective $\Pi = (\Gamma, \Omega, \Lambda)$ is defined as follows:

– Γ has scientific papers. In order to identify properties, predicates and relevant objects in the table we have used a description based on a RDF Schema and the *Science* Ontology.

– Ω are all the papers X_i satisfying the predicate TalkAbout(X_i,Biology).
– $\Omega^{'}$ has all the papers X_i added beacuse it satisfies the sentences in Λ.
– Λ has the transitive clausure of the hyponim relation in the Biology subtree.

In this way we produce an index for each word in the subtree using an RDF Schema and the science ontology to clarify the meaning of table names in the original database. This is the annotation process at the *perspective 0* level. Then we use the hyponym relations to add relations between words in the index, corresponding to our second level of annotation *perspective 1*.

Once the perspective is represented by an index (or by any other data structure implemented by the user) later queries take a considerably shorter time. In other words, from the point of view of performance, the *perspective 0* creation could take a long time, depending on the size of the database and the kind of processing performed on the raw data. However, once created, the annotations and indexes will speed up further processing, such as *perspective 1* creation and later queries and conjecture validations. In our example, such queries to the paper database take a time in the order of a few milliseconds, when executed from a remote computer located in the same local area network.

5 Related Work

Semantic techniques on grid environments can be roughly classified into two groups: those that provide knowledge about the grid resources and those that provide knowledge about the data grid contents [6]. The first one is used to describe, discover, manipulate and compose services while the second one is used to produce more knowledge through ontological resources in order to describe and discover new data relationships. In [7] a general architecture is proposed, in which there is a clear separation between the semantic grid level and the knowledge grid level. The semantic grid level uses ontologies to describe services in the grid while the knowledge grid level uses semantic techniques to process data and produce knowledge. Some of these proposals are based on computer agents [8] which can offer autonomy and negotiation capabilities to grid environments [9]

The Semantic Grid research community is mainly working on developing techniques using ontologies in order to improve knowledge access and recovery in the grid [10][11][12][13][14]. Ontology languages and reasoning techniques are fundamental to describe resources and services in this framework [15][16]. Most of the languages being considered use description logic to provide an automatic classification of resources and services with a model theoretic semantic. Recently, some proposals account for the lack of nonmonotonic reasoning techniques and rule languages usage in order to implement some of the requirements of the semantic web and semantic grid communities (for example negotiation of services) [17]. A main concern is to provide the adequate level of expressivity without loosing decidibility or tractability. The capability to describe resources

and services in a declarative language helps us to create automatic discovering and composition techniques which could improve the current capability of the grid to produce new knowledge.

The Virtual Data System [18] is an architecture for data virtualization. Using the virtual data language *VDL* users can describe workflows over datasets. Data transformation processes could be discovered and composed. Metadata about transformations, derivations, and datasets are registred in the distributed virtual data catalog. The *Knowledge Grid* [19] is an architecture for distributed data mining. The system uses ontologies [20] to describe data mining services and help users to elaborate data mining workflows. Comb-e-chem [21] is creating the infrastructure to analyze correlations and predict properties in chemical structures using techniques known as publication at source. Comb-e-chem provides services to create workflows, aggregate experimental data, select datasets and also annotate and edit data sources. Using the concept of *publication at source* all these data can be reused many times. MyGrid [22] offers an infrastructure to support research in bioinformatic. MyGrid provides data and resource integration services using semantic technologies to improve service discovery, data flow and distributed processing. Comparatively our proposal offers:

- A technique to link logical theories, described using FOL and Description Logics with data sources. This link explicitly shows relations among theories and data subsets producing indexes. These indexes improve data access in large datasets.
- Facilities to use a high level language (FOL) to describe data processing in data grids. Our data modeling process is completely defined with reference to FOL sentences. Annotation methods required to make *Perspective 0* annotations could be provided as libraries. In this way a researcher needs only to define the process by using FOL.
- A processing technique which leaves the data source unchanged.
- A flexible way to create views over data. Each user could have her own perspective over each data set.
- A process to identify objects, properties and relations in the framework of an arbitrary, user defined, theory. In this way the researcher could identify data objects confirming the theory used to process it.
- A technique for processing data at the source, avoiding issues related to the transfer of large amounts of data.
- An architecture of ontology services to implement the knowledge perspective concept.
- A technique to provide many points of view over data, increasing opportunities of knowledge discovery and scientific advance.

This is achieved through the combination of (1) a methodology based on ontology specification and knowledge representation and (2) appropriate data grid services that allow users to define their own ontological services.

6 Conclusions and Future Work

In this work we propose a methodology that establishes a bridge between data manipulation techniques based on ontological criteria and secure data access in grids. We base this methodology in a concept we call *Knowledge Perspective* which allows researchers to manipulate scientific data according to a theoretical framework.

From the viewpoint of knowledge representation and management, we propose the use of a high level language (First Order Logic) and a specification about how to compute a knowledge perspective. Using the grid environment each user could have the authorization level and enough computational and data resources to create indexes in the data source. We present a Globus-enabled Java platform that allows the users to define their own data services based on ontological description of the data. Both contributions allow the grid users to define new services and data access interfaces, consistent with their own knowledge perspectives.

Our initial results, reported in this paper, show the feasibility of using this concept when applied to frameworks where the information has low complexity levels. We need further research and tests for larger and more complex datasets. We describe distributed data sources using ontologies, facilitating data mediation and integrated access to heterogeneus data sources. We plan to implement further mediation techniques in the future. Ongoing research is oriented to applying and evaluating this technology in databases where the data objects are more complex, such as images. In this case the predicates associated to the objects can be satisfied using image processing algorithms.

References

1. Hey, T., Trefethen, A.: "e-science and its implications". Philosophical Transactions of the Royal Society **361**(1809) (2003) 1809–1825
2. Blanco, E., Cardinale, Y., Figueira, C., Hernndez, E., Rivas, R., Rukoz, M.: Remote data service installation on a grid-enabled java platform. In: 17th International Symposium on Computer Architecture and High Performance Computing SBAC-2005. (2005) 85–91
3. Cardinale, Y., Hernández, E.: Parallel Checkpointing on a Grid-enabled Java Platform. Lecture Notes in Computer Science (European Grid Conference EGC2005) (2005) To appear.
4. Fellbaum, C.: Wordnet: An Electronic Lexical Database. MIT Press (1999)
5. Freitas, F.: Ontology of science. Technical report, Universidade Federal de Santa Catarina (2001)
6. Goble, C., De Roure, D., Shadbolt, N., Fernandes, A.: Enhancing services and applications with knowledge and semantics. In Foster, I., Kesselman, C., eds.: The Grid 2: Blueprint for a New Computing Infrastructure. Morgan-Kaufmann (2004)

7. Goble, C., De Roure, D.: The semantic grid: Myth busting and bridge building. In: 16th European Conference on Artificial Intelligence (ECAI-2004), Valencia, Spain (2004) 1129–1135
8. Rana, O.F., Pouchard, L.: Agent based semantic grids: Research issues and challenges. Journal of Parallel and Distributed Computing Practices (2003)
9. Roure, D.D., Shadbolt, N., Jennings, N.: The semantic grid: Past, present and futur. In: Proceedings of The IEEE. (2005)
10. Goble, C., De Roure, D.: The semantic web and grid computing. In Kashyap, V., Shklar, L., eds.: Real World Semantic Web Applications. Volume 92 of Frontiers in Artificial Intelligence and Applications. IOS Press (2002)
11. Cannataro, M., Talia, D.: Semantics and knowledge grids: Building the next-generation grid. IEEE Intelligent Systems **19**(1) (2004) 56–63
12. Chen, L., Shadbolt, N., Tao, F., Puleston, C., Goble, C., Cox, S.: Exploiting semantics for e-science on the semantic grid. In: Web Intelligence (WI2003) workshop on Knowledge Grid and Grid Intelligence. (2003) 122–132
13. De Roure, D., Hendler, J.: E-science: the grid and the semantic web. IEEE Intelligent Systems **19**(1) (2004) 65–71
14. Newhouse, S., Mayer, S., Furmento, S., McGough, S., Stanton, J., Darlington, J.: Laying the foundations for the semantic grid. In: AISB Workshop on AI and Grid Computing. (2002)
15. Horrocks, I.: Daml-oil: A reason-able web ontology language. In: Proceedings of EDBT. Number 2287 in Lecture Notes in Computer Science, Springer (2002) 2–13
16. et al, M.D.: Owl: Web ontology language 1.0 reference. Technical report, World Wide Web Consortium (2002)
17. Kifer, M., Bruijn, J.d., Boley, H., Fensel, D.: A realistic architecture for the semantic web. In: International Conference on Rules and Rule Markup Languages for the Semantic Web. (2005)
18. Foster, I., Voeckler, J., Wilde, M., Zhao, Y.: The virtual data grid: A new model and architecture for data-intensive collaboration. In: CIDR 2003 Conference on Innovative Data System Research. (2003)
19. Cannataro, M., Talia, D.: The knowledge grid. CACM **46**(1) (2003) 89–93
20. Cannataro, M., Comito, C.: A data mining ontology for grid programming. In: 1st International Workshop on Semantics in Peer-to-Peer and Grid Computing (SemPGrid2003). (2003)
21. Frey, J.G., Bradley, M., Essex, J., Hursthouse, M., Lewis, S., Luck, M., Moreau, L., De Roure, D., Surridge, M., Welsh, A.: Combinatorial chemistry and the grid. In Berman, F., Hey, A.J., Fox, G.C., eds.: Grid computing: making the global infrastructure a reality. Wiley Series in Communications Networking and Distributed Systems. John Wiley & Sons Ltd., Chichester, UK (2003) 945–962
22. Goble, C., Pettifer, S., Stevens, R., Greenhalgh, C.: Knowledge integration: In silico experiments in bioinformatics. In Foster, I., Kesselman, C., eds.: The Grid: Blueprint for a New Computing Infrastructure Second Edition. Morgan Kaufman (2004)