COMPUTER SCIENCE LETTERS

www.csl.issres.net Vol. 1 (1) – June 2009

A Semantic Description and Registration Framework for Large Grid Resource Discovery Systems

Mahamat Issa Hassan[1] and Azween Abdullah[2]
[1,2] Department of Computer and Information Sciences
Universiti Teknologi PETRONAS, Bandar Seri Iskandar, 31750 Tronoh, Perak, MALAYSIA

Abstract

Large grid systems are characterized by the scale of resources and users associated with high heterogeneity and dynamism, which entails a resource discovery (RD) system that is able to cope with this condition. This paper proposes a novel semantic description and registration framework for large grid RD systems. The description model consists of a set ontologies and services. Ontologies are used as a data model for resource and application description, whereas the services are to accomplish the description process. The registration model is also based on ontology, where nodes of the resources are classified to some classes according to the ontology concepts, which means each class represents a concept in the ontology. Each class has a head, which plays the role of a registry in its class and communicates with the other heads of the classes during the discovery process. The paper shows how the framework satisfies grid RD features such as scalability, decentralization, dynamism and interoperability.

Keywords: Service discovery; grid computing; peer-to-peer networks; and ontology.

1. Introduction

Grids enable aggregation and sharing of geographically/Internet-wide distributed heterogeneous resources such as computers, databases, visualization devices, and scientific instruments [1, 22]. *Large grid system* is a multinstitutional based grid system that accommodates large number resource nodes and users, or in other words, is a federation of small grids into a global one [2]. Grid RD refers to the process of locating suitable resources based on user requests [3]. RD system should satisfy both the resource *provider* and *consumer* by allowing the *provider* to describe his/her resources and *consumer* to prepare his/her query and searches on the network to get the relevant resources from the described resources.

RD system represents an important component in the grid system, as resource reservation and task scheduling are based on it to enable grid applications development. Unfortunately, large grid systems are normally associated with some complexities such as: (i) resources and users are distributed across different locations; (ii) resources (e.g., CPU, network and storage) are heterogeneous in their platforms; (iii) statuses of the resources are dynamic (resources can join or leave the system without any prior notice) when they are registered in the registration pool(s); and (iv) grids are often distributed across security domains with large number of resources involved [7-9]. These complexities have raised several requirements that should be addressed by any developed RD system. These requirements include *decentralization*, *scalability*, *dynamism*, and *interoperability*. Accordingly, a grid RD system should be fully decentralized from any global control, tolerates intermittent resource participation (either voluntary or due to failure) [10], performs as expected regardless of any changes on the quantity of users or resources; and supports semantic description for resources and applications [11]. As a result, it is challenging indeed to develop efficient system to discover the dynamic resources.

This paper proposes a novel semantic description and registration framework for large grid RD systems. The description model uses ontologies as a data model for resource and application description, and the registration model also uses ontology to construct the architecture of the grid nodes.

The following sections comprise the paper: Section 2 describes a generic RD system and related work, the proposed description and registration framework are presented in section 3, section 4 presents an application, section 5 discusses pertinent issues, and section 6 concludes the paper.

2. A Generic Grid RD System and Related Work

In order to understand the related work and the proposed RD system, we present a generic RD system, which is discussed in the coming subsection followed by the related work.

2.1. The Generic Grid RD System

A grid RD system should contain three components, namely *description*, *registration* and *locating* (which is composed of *search* and *matching)* (see figure 1). *Description* refers to the abstract representation of a resource nature and its capabilities. This abstract representation allows consumer to use/invoke the resource, as it is the only means to justify the existence of the resource. *Registration* is related to *where* to store the resource information[1] (metadata) and *how long* to keep this information. Note that, the resource information in the registration component is presented in a resource description format,

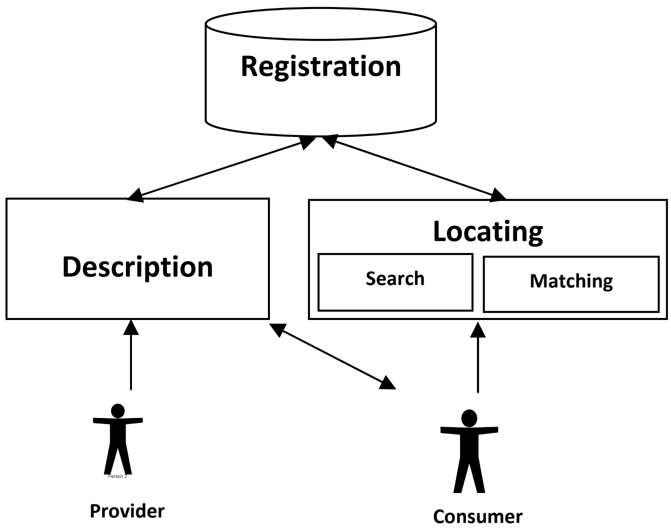

Figure 1. The generic grid RD system

[1] We use the terms resource information, information of the resource(s) and metadata interchangeably.

which is provided by the *description* component. Locating component further consists of two subcomponents: *search* and *matching*. The former is about how to route a resource request from the consumer node to the registration node(s). Meanwhile, the latter is about how to evaluate the metadata that are located in the registration with regard to a resource requests. The interaction between these components can be captured through how a user *(provider/consumer)* can invoke the RD system. A "*user*" here, can be an actual end user or a software agent that functions on. The provider uses the description mechanism to describe its resources in a format that is suitable for registration and matching. The description of the resource is then registered at the registration pool. Once the resource is registered, the *consumer* uses the description mechanism to specify the capabilities/features of its requested resource. The *search* algorithm *takes* the resource request and searches on the available registration pool(s). The matching mechanism compares the user request to the available resource information and returns all the resources that fulfill the request specifications.

2.2. Related work

Currently, there are two types of registration and description mechanisms in the grid RD systems. RD registration models can be *centralized* or *hierarchical*. In the former, the metadata is indexed under a centralized node, and users send their resource queries to that node. The resource providers update their resource status at periodic intervals using resource update messages. Condor system [12] is an example of the centralized systems. In Condor model, the centralized node is called Central Manager (CM), which collects information about the status of the resources from the resource providers. The resource providers are represented by Resource-owner Agent (RA), which is located in each resource provider. The CM then, receives users' tasks and matches them with the resources. In hierarchical models however, the metadata is indexed under a set of nodes in a hierarchical manner (each chilled node indexes its metadata on its parent node, which means the root node will have the entire metadata of the system). The *monitoring and discovery service* (MDS) of Globus [13] implements this model. It uses two services: a configurable information provider called *grid resource information service* (GRIS) and a configurable aggregate directory service called *grid index information service* (GIIS). A GRIS answers queries about the resources of a particular node. A GIIS combines the information provided by a set of GRIS services managed by a given Virtual Organization (VO).

Both of the models (centralized & hierarchical) have some issues with regard to the large grid RD requirements [14, 21, 22]. For example, in Condor system, the Central Manager that matches the resources with the users' tasks may be a point of failure. In Globus MDS, the updates on GRISs at the lowest levels do not automatically propagate up to the top of the hierarchy, which means the available resource information may not be completely up-to-date. This has motivated researches recently to focus on peer-to-peer based registration models, in which no node is important than the others. P2P based registration can also be divided into structured and unstructured. In the former, resource information is indexed into a set of nodes and users query these node based on the resource attributes. For example, one node may be responsible about CPU and memory resources [17-19]. Meanwhile, in the latter, each node indexes its resource information locally, and requests and responses are forwarded with a hop-by-hop mechanism from and to the request origin [20]. The P2P based registration models also have some shortcomings. For instance, the structured registration models have high network maintenance, which results from the frequent updates of resources' status. While unstructured registration is associated with high network traffic as they use hop-to-hop mechanisms. For this reason, several techniques are proposed to reduce the traffic, and yet they produce low precision during the discovery process.

Regarding the existing resource description models, all but the current RD systems that have been discussed above, use keyword based indexing data models such as bit map, directories, and relational databases to describe their resources. Keyword based indexing may not be able to cope with multinstitutional participants of large grids. For example, a requested resource might be semantically similar to an available described resource, but syntactically different (synonyms); or a requested resource might be syntactically equivalent to an available described resource, but semantically different (homonyms).

More recently, there have been few studies that propose the semantic description to the RD systems. They introduce ontologies as a data model for resource description and request. Nevertheless, these models are mainly built based on the existing registration model, which may no longer be suitable for large grids. For example the work of [23] is based on the centralized systems, [15] is for the hierarchical ones, and [14] is a pure agent-based unstructured P2P. It should be noted that meeting large grid requirements depends on how the RD components are modeled. For example, having an expressive resource description mechanism eases the matching process between resource request and metadata, and also enhances the precision, which in turn meets interoperability. Good registration pool architecture associated with a dynamic information update mechanism provides scalability, decentralization and dynamism to the system.

To that effect, we propose a novel semantic-based description and registration framework. Our work aims at contributing to the development of a sophisticated RD system that takes into account the identified large grid requirements, and at providing a scientific progress further than the state-of-the-art in this field. In contrast to most other works, we introduce ontology as a data model for describing the resources and applications, a semantic network architecture for resource registration. Our framework provides an ontology based model that can satisfy interoperability and a class based node organization, in which nodes of the resources are semantically grouped to allow them to index their resource information in a decentralized and scalable manner. In short, with this framework, we are solving the shortcomings of both the existing grid RD systems and the research oriented P2P based grid RD systems.

3. The Proposed Description and Registration Framework

3.1. The Description Model

We build our description by considering two factors, which are: *easy* and *expressive* resource description and *easy* resource request formulation. To achieve that, we propose the ontologies as data model for resource description and request, and implement the concept of *selling* and *buying* product process of our daily live as a scenario for describing and requesting the resources. Ontology is defined as the formal specification of a vocabulary of concepts and axioms relating to them [4, 5]. It formally specifies how to represent objects, concepts and other entities that are assumed to exist in some area of interest and the relationships among them [6]. The concept of *selling* and *buying product* normally as follows: a seller advertises his/her product as wide as possible using the available standard advertising tools (e.g. news paper). The seller also wants to get a feedback about the product demand by using the existing techniques (e.g. data warehouse). On the other hand, the buyer must have a general product goal (e.g. computer) that he/she wants to buy. The buyer further narrows down his/her goal to goal specification (e.g. computer brand, memory size, CPU frequency, and so forth) upon which the choice may be made. When the above steps have taken place, then a transaction between the seller and the buyer may be accomplished. In short, the proposed description is composed of two ontologies and three other components. The ontologies are named *resource ontology* and *application ontology*, whereas the other three components are *semantic description manager*, *semantic metadata repository*, and *semantic metadata summary pool*. The details of these elements are described in the following subsections.

3.1.1 Semantic Description Manager (SDM)

SDM is generally responsible for the resources and applications description data model. It consists of the two ontologies (the *application* and *resource*), and *Visualizer and wrapper*. Application ontology describes all the existing and expected applications for the grid system. The description should provide all the essential information about applications such as their domains, required resources, their relations with each other and so on. Resource ontology is similar to the application one, but is more towards the resources. It describes the resources' attributes, the relation among the resources, the provided service and so on. Both of these ontologies can be instantiated by the user. For example, the provider may instantiate the resource ontology and consumer may instantiate the application ontology. Visualizer and wrapper have two tasks. First, it visualizes the ontology in a graphical format so that a user can simply see all the defined resources and applications for the system. Second, it copes with language and interface differences between the user node and the assigned SDM node. Remember, in large grids we federate small grids into a global one, in which is not practical to impose all the participants in using one language or interface.

3.1.2 Semantic Metadata Repository (SMR)

SMR instantiates the ontologies of SDM and stores the instances of the resources and applications with their actual attribute values in some local repositories. Every node (physical organization/sm

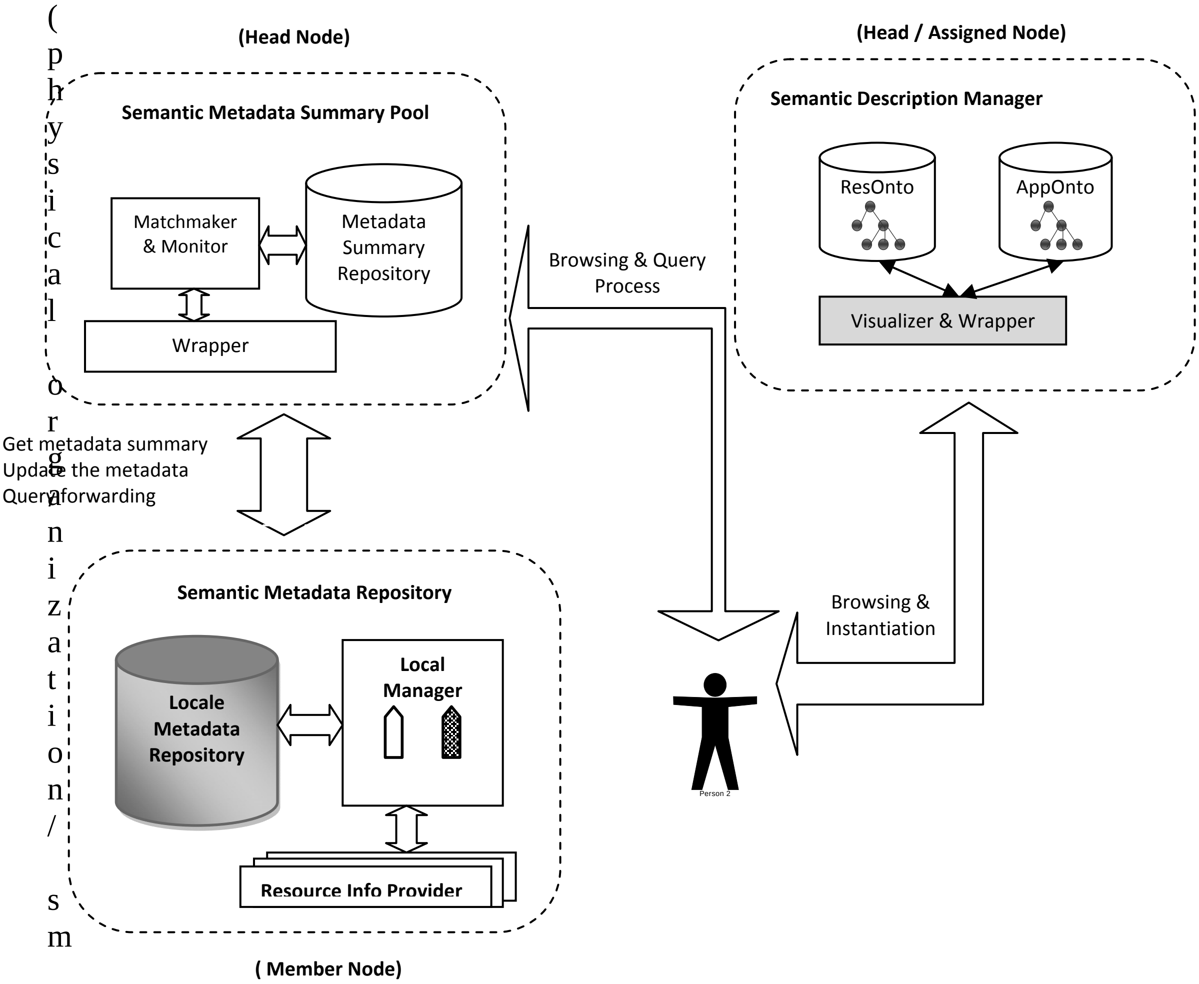

Figure 2. The proposed description model

all grid) may have one SMR. SMR has three components: *local manager*, *local metadata repository* and *resource info provider*. Local manger aggregates and updates the information of the local resources and applications, and other resource information that is received from an external node if it is assigned to. The local manager also provides a summary of its local repository to the *semantic metadata summary pool* and handles the resource queries when they are sent. The local metadata resource repository is where to gather resource information. Information provider can be simple centralized information system that provides information about some resources.

3.1.3 Semantic Metadata Summary Pool (SMSP)

The SMSP acts as a registry for the resource and application information summaries that are provided by the different SMRs. It is composed of a repository, matchmaker & monitor and wrapper. The repository stores the aggregated information summaries and provides a browsing functionality for these summaries. The monitor updates the summaries when there is a change, and handles the initial preprocess of the resource queries. The resource query preprocess includes the initial matching of the request against the summaries and forwarding the query to the relevant SMR based on the initial matching result and so on.

3.2. The Registration Model

We propose here a semantic class based architecture for the resource and application registration (registry). In this architecture, nodes are organized into some classes. Each class will have a head, which is selected among the other nodes. The classification criteria and the number of classes are done by the use of *ontology*. We name this ontology *dictionary ontology (DicOnto)* (see figure 3). Formally, DicOnt can be seen as a collection of three things: *concepts* (C), *properties* (P), and *relationship* between those concepts (R). Where, C is a set of grid resource/application concepts, P is the set properties of the concepts and R is the relations between the resource concepts, which can produce the concept hierarchy (tree structure). For example a *computer* and *operating system* can represent two concepts, the relation between these concepts is that, a computer has an operating system, and the property of the concept operating system is its version (Windows, Linux, Apple).

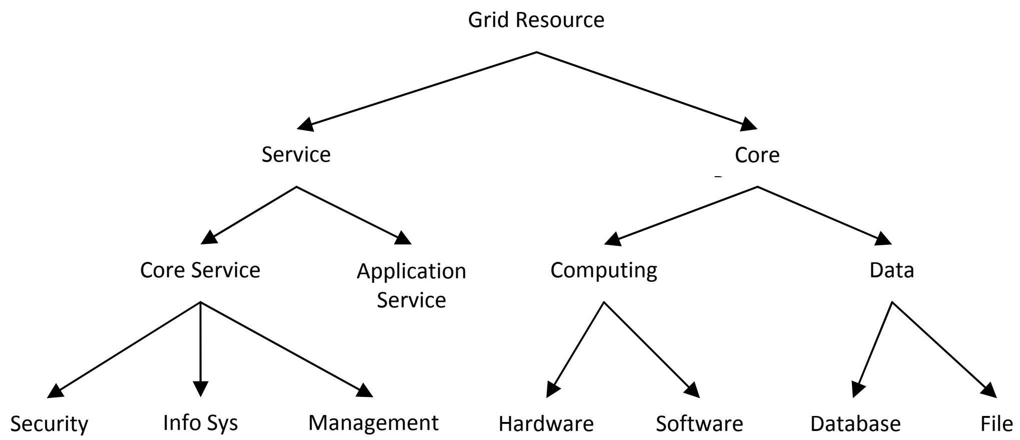

Figure 3. The dictionary ontology

In order to have a clear picture about the integration of DicOnt in building the class based architecture, we define the architecture components as follows:

Node: a grid node that provides one or a set of resources that are clearly identified in the dictionary ontology. Node represents a physical organization such as a university, research center, weather center, and so on.

Class: a collection of nodes that their resources are the instances of a particular concept in the dictionary ontology. For example if we take the *computing* resource in figure 3 as

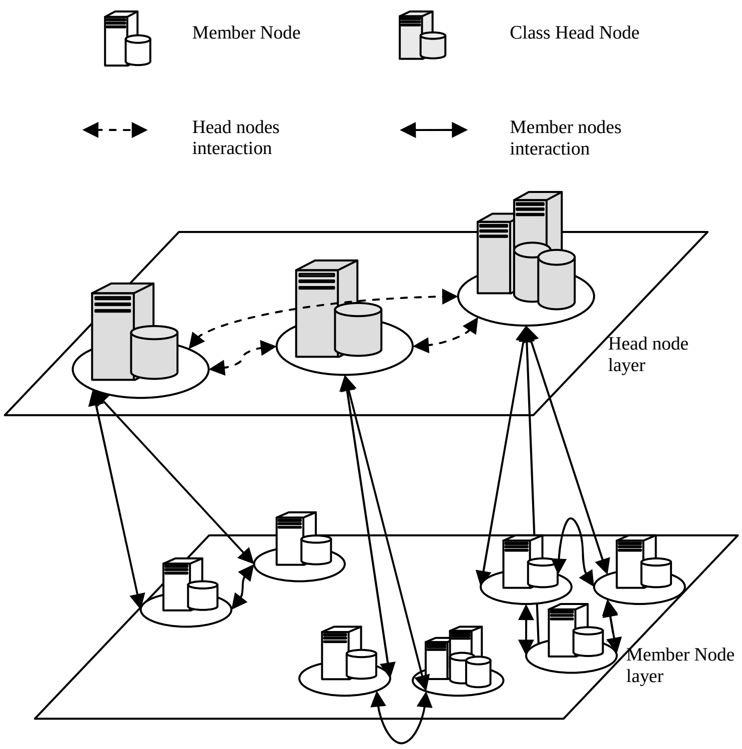

Figure 4. The proposed RD registration architecture

a concept, the class of the computing resources will include all the nodes that have software and hardware resources.

In addition to that, a class is supposed to have two kinds of nodes namely *member* and *head*.

Member: is a node that belongs to a particular class based on the resource concept instanceship.

Head: is a selected node from the member nodes of a given class. It should have the highest rank to meet some clearly defined performance capabilities and availability. Performance capabilities such as the speed of the server, network bandwidth, and reserved memory space for resource information, and so on. Availability is the proportion of time when node is persistent in a grid system. Grid nodes are very dynamic due to some circumstances such as local loading and network problems. The construction of the classes and assignment of the heads can be accomplished through some sets of processes and algorithms which can be found in our work of [25]. Figure 4 describes the overall registration architecture.

4. Application

In order to clarify the interactions between the new description and the registration components during the resources discovery process, we introduce an example that shows how a grid user can describe or request a resource (description), and how resources can be discovered (discovery).

Assuming that we build a grid with some nodes distributed in different locations. Using the DicOnt model of section 3.2, we can define the resource concepts, each concept may contain some specific resources. Thus, implementing the class based architecture, we obtain a set of classes with their heads. Each class represents exactly one concept at the DicOnt. Each node (head/member) has a SMR, and each head has SMSP. Each class has a SDM which is located either in the head of the class or any assigned node within the class. First we will show the description steps, followed by the discovery.

4.1. Description

Step 1: The SMR of the provider node instantiates the resource ontology of SDM which belongs to its class.

Step 2: The SMR adds the attribute values of its resources and stores them in its repository. It then summarizes the stored resource information and sends that summary to both its head (SMSP) and all the neighboring node members (SMRs). Note that the difference between the resource information and their summary is that, resource information contains all the attributes that describe the resource, which are formally stated in the resource ontology. Whereas, the summary is just the name of the resource or a combination of the name with other attributes that can show the existence of the resource in the node.

Step 3: In case of any changes on resource information, the SMR updates its neighbors SMRs as well as its head SMSP.

The result of these steps is that, each member node (SMR) has a summary of its neighbors and each head (SMSP) the overall summary of the class.

4.2. Discovery

We now turn towards the resource request formulation and the look up. In this case, we assume that the user is belonging to a particular node (SMR).

Step 1: The user may browse the application ontology and instantiates the application ontology. This application instance will include all the required resources for that application. For example, a *clustering* application may require the data that need to be clustered, clustering method, CPUs, and so on.

Step 2: The user then adds the quantitative and qualitative attribute values for the required resources of the application, by doing so the application and its required resources form the request format.

Step 3: The user node checks its aggregated resources summaries, which are provided by its neighbors, if there is an initial match between the request and the summaries. The node sends the request to the relevant nodes that are associated with all or part of the requested resources.

Step 4: If there is no answer for the request from the neighbors or there are still reaming some requested resources, the SMR sends the request to the head node/SMSP. Subsequently, the SMSP does the initial matching with the aggregated summaries. If there is an initial match between the requested resources and the summaries, the SMSP forwards the request to the relevant nodes in the class and forwards back the answers to the request origin, if any.

Step 5: If the there is no initial match with summaries, which indicates the requested resources are not available in this class, the SMSP forwards the request to the relevant SMSP(s). The relevancy of SMSP for which a request is to be forwarded is based on requested resources concepts, which is stated in the DicOnt. Therefore, each SMSP that receives a request will perform the same functions of initial matching and request, and answer forwarding of step 4 until all the resources are found.

5. Discussion

The most important requirements of large grid RD systems are scalability and interoperability due to the large number of resources and users as well as the high rate of heterogeneity among the participants. In this section, we discuss how the proposed framework meets interoperability and scalability. *Interoperability* is the ability of a RD system to span multiple administrative domains in discovering the resources. The use of ontology in this framework allows a well defined meaning to resource information and provides a uniform description, and discovery semantic among the participants. There will be no syntactic matching as in the case of most of the current systems. *Scalability* in grid is the increase of the number of resources or users that use the resources. Our framework is scalable by using the semantic class based node architecture, which gives an opportunity to any class to grow upwards according to the DicOnt. This means, the minimum number of the class is one, thus, the overall resources of the system are treated as one concept, and the maximum number of the classes is total number of the resources when each individual resource is treated as a concept in the DicOnt. Therefore, the system can grow between the two cases. Meanwhile, the role of the heads in the classes and the interclass communication shows that the framework also meets decentralization and dynamism features.

6. Conclusion

In this paper, a novel description and registration framework for large grid RD systems is presented. The framework consists of two aspects: description of the resources and registration of the described resource. The paper proposes ontology model as a data model for describing the resources, applications and their relationships, and set services to accomplish the description process. Semantic registration architecture is also presented. In this case, ontology is used to arrange the resource to different types. Each type represents a resource concept on the ontology. The concepts are mapped to a class based architecture (each concept is a class of nodes). Each class has a head that represents a resources and application registry, and eases the communication with other heads. The scope of the resource look up starts by the class of query origin first, then to the other class. We present an application that shows how the framework handles the resources discovery process. Through the discussion, we have shown that the framework satisfies the large grid RD requirements, which are scalability, decentralizations, dynamism, and interoperability.

References

[1] Foster I., and C. Kesselman, *The Grid 2: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, 2003.

[2] Kurdi, H., M. Li, and H. Al-Raweshidy A Classification of Emerging and Traditional Grid Systems. *IEEE Distributed Systems Online 9*(3). 1, 2008.

[3] Iamnitchi, A., and I. Foster On Fully Decentralized Resource Discovery in Grid Environments. *Lecture Notes In Computer Science 2242*. 51-62, 2001.

[4] Chandrasekaran, B., J. R. Josephson, and V. R. Benjamins What are ontologies, and why do we need them? *IEEE Intelligent Systems*, *14*. 20-26, 1999.

[5] Lacasta, J., J. Nogueras-Iso, R. Be´jar, P.R. Muro-Medrano, F.J. Zarazaga-Soria A Web Ontology Service to facilitate interoperability within a Spatial Data Infrastructure: Applicability to discovery. *Data & Knowledge Engineering*, *63* (3). 947-971, 2007.

[6] Trunfioa, P., D. Taliaa, H. Papadakisb, P. Fragopouloub, M. Mordacchinic, M. Pennanend, K. Popove, V. Vlassovf, and S. Haridi Peer-to-Peer resource discovery in Grids: Models and systems. *Future Generation Computer Systems*, *23* (7). 864–878, 2007.

[7] Padmanabhan, A., S. Wang, S. Ghosh, and R. Briggs. A Self-Organized Grouping (SOG) Method for Efficient Grid Resource Discovery *Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing*, IEEE Computer Society 2005, 312-317.

[8] Hassan, M., and A. Abdulah. Self-organizing Grid resource discovery *International Symposium on Information Technology (ITSim 2008)*, IEEE Xplore, Kuala Lumpure-Malaysia, 2008, 1-4.

[9] Hassan, M., and A. Abdulah. Scalable Self-Organizing Model for Grid Resource Discovery *International Conference on Network Applications, Protocols and Services 2008*, Kedah Darul Aman-Malaysia, 2008.

[10] Padmanabhan, A. SOG: A Self-Organized Grouping Infrastructure for Grid Resource discovery *Graduate College*, University of Iowa, Iowa, 2006, 5.

[11] Hassan, M., and A. Abdulah. Semantic- Based Scalable Decentralized Grid Resource Discovery *The 2009 International Conference on e-Technology (e-Tech 2009),* Singapore, 2009, 3316-3324

[12] Condor Project: http://www.cs.wisc.edu/condor/

[13] Globus Toolkit: http://www.globus.org.

[14] Han, L., and D. Berry Semantic-supported and agent-based decentralized grid resource discovery. *Future Generation Computer Systems*, *24* (8). 806-812, 2008.

[15] Said, M.P., and I Kojima S-MDS: Semantic Monitoring and Discovery System. *Journal of Grid Computing 7*(2). 205-224, 2009.

[16] Somasundaram, T.S., R.A. Balachandar, V. Kandasamy, R. Buyya, R. Raman, N. Mohanram, and S. Varun. Semantic-based Grid Resource Discovery and its Integration with the Grid Service Broker *International Conference on Advanced Computing and Communications (ADCOM 2006)* 2006, 84-89.

[17] Andrzejak A., and Z. Xu, "Scalable, Efficient Range Queries for Grid Information Services", *in Proceedings of the Second International Conference on Peer-to-Peer Computing (P2P 2002)*, pp. 33-40, 2002.

[18] Bharambe, A.R., M. Agrawal, and S. Seshan. Mercury: supporting scalable multi-attribute range queries *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, Portland, Oregon, USA, 2004, 353-366.

[19] Cai M., M. Frank, J. Chen and P. Szekely, "MAAN: A Multi-Attribute Addressable Network for Grid Information Services", in *Proceedings of 4th International Workshop on Grid Computing (GRID 2003)*, pp. 184-191, 2003.

[20] Iamnitchi, A., and I. Foster On Fully Decentralized Resource Discovery in Grid Environments. *Lecture Notes In Computer Science 2242*. 51-62, 2001.

[21] Mastroianni, C., D. Talia, and O. Verta Designing an information system for Grids: Comparing hierarchical, decentralized P2P and super-peer models. *Parallel Computing*, *34* (10). 593-611, 2008.

[22] Ranjan, R., A. Harwood, And R. Buyya Peer-to-Peer Based Resource Discovery in Global Grids A Tutorial *IEEE Communications Surveys & Tutorials*, *10* (2). 6-33, 2008.

[23] Asadzadeh, P., R. Buyya, C. Ling Kei, D. Nayar, and S. Venugopal, "Global Grids and Software Toolkits: A Study of Four Grid Middleware Technologies," *High Performance Computing: Paradigm and Infrastructure*, L. Y. a. M. Guo, ed., New Jersey, USA: Wiley Press 2005

[24] Ludwig, S.A., and S. M. S. Reyhani Introduction of semantic matchmaking to grid computing. *Journal of Parallel and Distributed Computing 65* (12). 1533-1541.

[25] Hassan, M.I., and A. Abdulah A New Resource Discovery Framwork. *The International Arab Journal of Information Technology (IAJIT), (under publication), 2009*.