

A Collaborative Inter-Data Grid Strong Semantic Model with Hybrid Namespace

Dalia El-Mansy

The Southern Methodist University/Department of Computer Science, Texas, USA

Email: dalia21@aucegypt.edu

Ahmed Sameh

The American University in Cairo/Department of Computer Science, Cairo, Egypt

Email: sameh@aucegypt.edu

Abstract— The Data Grid, like all other collaboration models, has strict rules for contributors to follow and many criteria to abide with. Namespace is one of the rules that govern the contributors. Some fields are not ready for abiding with such kinds of global rules. For instance, scientific research taxonomy (down to topics and areas of interest) is highly dynamic. Topics are confusingly interdisciplinary and sometimes have institutional influence. The intention is to design a hybrid namespace collaboration model for existing organizations to pay lower cost (initial and running) in order to join it. Data resources will be allowed to have colliding names. The contributing organization will be free to set any rules for its internal users to follow (such as taxonomy). However, it will have to take care of a very simple interface to the intended model in order to introduce new contributed resources or to query external ones. A hierarchical hybrid namespace will be maintained in order to uniquely identify data resources that have colliding names. Similar and related topics in different places in the hierarchy will be linked by semantic relations to express the degree of similarity and or relevance. Resource explorations should follow the suitable semantic relations while traversing the hierarchy of servers and data grids for more successful search.

Index Terms- Data Grids, Semantic Taxonomy, Hybrid Namespace

I. INTRODUCTION

A. Preview

In an increasing number of scientific disciplines, large data collections are emerging as important community resources. In domains as diverse as global climate change, high-energy physics, and computational genomics, the volume of interesting data is already measured in terabytes and will soon total petabytes. The communities of researchers that need to access and analyze this data (often using sophisticated and computationally expensive techniques) are often large and are almost always geographically distributed, as are the computing and storage resources that these communities rely upon to store and analyze their data.

This combination of large dataset size, geographic distribution of users and resources, and computationally

intensive analysis results in complex and stringent performance demands that are not satisfied by any existing data management infrastructure. A large scientific collaboration may generate many queries, each involving access to, or supercomputer-class computations of gigabytes or terabytes of data. Efficient and reliable execution of these queries may require careful management of terabyte caches, gigabit/s data transfer over wide area networks, co-scheduling of data transfers and supercomputer computation, accurate performance estimations to guide the selection of dataset replicas, and other advanced techniques that collectively maximize use of scarce storage, networking, and computing resources.

Next generation scientific exploration requires computing power and storage that no single institution alone is able to afford. Additionally, easy access to distributed data is required to improve the sharing of results by scientific communities spread around the world [5]. The proposed solution to these challenges is to enable different institutions, working in the same scientific field, to put their computing, storage and data resources together in order to achieve the required performance and scale.

Obviously this is not an easy task. Researchers need to access all of the resources in a uniform, transparent, and easy way and many challenges have to be handled to achieve this goal. Different institutions may use different computing and storage systems and will also have local security rules.

Data Grids have been active area of research since the early 90's. The aim is to provide solutions for resource location transparency and massive data transfers. A Grid allows its constituent resources to be used in a coordinated fashion to deliver various qualities of service, relating for example to response time, throughput, availability, and security, and/or co-allocation of multiple resource types to meet complex user demands, so that the utility of the combined system is significantly greater than that of the sum of its parts. [5].

A data grid is a grid computing system that deals with data — the controlled sharing and management of large amounts of distributed data. These are often, but not

always, combined with computational grid computing systems [14].

B. Problem Definition

The Data Grid functional design assumes uniformity of information infrastructure. This uniformity is intended to sharpen the collaboration operability. However, it narrows the scope of usability by missing possible cooperation opportunities.

Despite the fact that it would look nicely flexible, the Data Grid could frustrate future joiners just because the architectural motivations break their vision (especially if the new comers had partial commonality in the data grid interest). The lack of provision on the role of interests' commonality makes the Data Grid fall short in handling reality. The example of scientific research institutions and their common research points is used in this research to highlight this issue.

Many scientific research fields show different degrees of resemblance. Researchers pay the effort for discovering and collecting suitable resources from different branches of science depending on their sense and common knowledge.

Also, researchers pay the effort of tracking out the aliasing maze caused by the different names the same branches of science are taking in different institutions and countries. Many opportunities of resource discovery must be lost unless an effective automation is achieved for the effort spent to discover resources around similar branches of scientific research. Also, a unified taxonomy, if imposed by a single data grid, will prevent all but those who are already using that taxonomy from contributing and benefiting.

C. Proposed Solution

In this research, a wider collaborative model is designed in an effort to solve the aliasing problem and the partial commonality issue. A middle ground solution is the one that lets the research centers in different institutions follow their preferred taxonomies while automating the effort the researchers pay when looking around for knowledge. Semantic relations are defined to be established between branches of scientific research to express the degree of commonality. Searches will be guided by those semantic links when traversing the huge community. Aliases will be linked by "Same" relations while partial resemblance will be expressed by "Similar" relations.

The rest of the paper is organized as follows: *Section 2* presents the collaboration model with hybrid namespace. *Section 3* presents the architecture of the model, *Section 4* presents implementation details, *Section 5* presents some experimental results, *Section 6* presents related work and *Section 7* is the conclusion.

II. A COLLABORATIVE MODEL WITH HYBRID NAMESPACE

A. Introducing the Model

1 World Before:

Namespace is the strictest rule that would frighten contributors away from a collaboration model. Usually,

the namespace, or any other globally enforced rule, is stored and maintained centrally to be available to all contributors.

Data Grid is decentralized by nature except for the namespace. It's kept and maintained centrally in all models to guarantee unique mapping from the set of names declared in the central namespace to the set of resources stored around the grid. The assumption behind this is that the effort spent to access the namespace is negligible as compared to the effort spent to access the actual referenced data. The current definition of Data Grids is structured around the idea of sharing extremely huge and geographically distributed data resources.

A typical example where contributors are unable to follow a unified taxonomy or naming system is the domain of scientific research. For instance, the word "Perception" is a research area in different sciences. It is used in Psychology, Artificial Intelligence and other disciplines almost in the same sense. There are other examples like the words "Cognition", "Memory"...etc.

Additionally, different sciences can share a significantly large number of topic names and areas of research but in completely different classifications. Examples for that are "Medicine", "Veterinary Medicine", "Pharmacology" and "Biology". For instance, the effect of hormones on the metabolism of nutrients can be found as a research area in any of those four sciences.

In real life, naming is a free-minded activity that always reflects the personality and intellectuality of the population. Another example for troubles with naming issues is the different classifications of branches and sub-branches of sciences accredited and followed by different research institutions.

A researcher can find his/her topic (or research area) classified completely different in another university or research center. For instance, Algorithms is classified under Computer Science in some universities and under Applied Mathematics in other universities.

This could explain the long absence of a unified structured model that can bind all research centers around the world in one collaborative database; The model that could have been easily realized if all the research centers around the world were using a unified science taxonomy (branches, sub-branches, research areas, topics...etc.).

For that reason, Data Grid projects are always isolated islands. Contributors do not have a decent way of knowing about each other. They do not recognize different degrees of similarity. Therefore, they do not benefit from it. They could miss cooperation opportunities just because they have different taxonomies. A wider collaborative model is needed to join contributors and data grids in a relaxed aggregation. Inter-data grid communications are needed.

The current situation is that each research center (or group of related ones in a bigger organization) has its own collaborative models that reflect its taxonomy of scientific branches and research topics. Hence, researchers from different research centers (different environments) need to do unstructured search into other models.

For example, Internet search engines use keywords for text content search. Consequently, the search results are sometimes unacceptably fuzzy and unmanageably huge.

2 World After:

To define the model more precisely, we start by designing a collaborative model for scientific research centers to collaborate all over the world as if they all use one hierarchical namespace for research topics (unified taxonomy). A hierarchy of servers will be built following the taxonomy. Level one is the major group of sciences (Humanities, Natural Sciences, Physical Sciences...etc.). Each group of science branches into the sciences of that group. So, we can have Mathematics, Physics, Chemistry, Computer Science...etc. under Physical Sciences. Pure Mathematics and Applied Mathematics will lie under Mathematics. Under Pure Mathematics, we can have Calculus, Algebra, Geometry and Topology. And so on until we have leaf fields for topics and research points.

It is noticed that scientific research classification is unified down to a certain level then varies from place to place. For instance, the groups of sciences are almost the same everywhere. Also, many sciences are branching from the same group of sciences everywhere as well (Chemistry, Physics, Mathematics are always branching from Physical Sciences group). The middle ground between the ideal case, that does not exist, and the existing case is to unify the upper part of the hierarchical namespace that is common worldwide. Then let research centers join on that basis and start their different classifications.

The common part of the namespace (the upper part) can be represented by a hierarchy of huge servers. Those servers can reside at any location. Many networking solutions can be applied to resolve bottlenecks depending on the assumption that those servers will be almost read only. Institutions' servers can branch from any level of the common namespace server hierarchy.

For example, "Computer Science" server of a given university can branch from the "Physical Sciences" server while "Computer Science" server of another university or research center can branch from the "Mathematics server". Under each of those two servers, a whole hierarchy of servers is built reflecting the taxonomy of Computer Science branches, sub-branches, research areas, and topics accredited by the corresponding university or research center. The common namespace servers are denoted as "Common Servers" while the underneath servers at different institutions are denoted as "Institutional Servers."

Equivalent servers at different institutions can be linked together (or to Common servers) through "Same" relations. The link is saved on both servers after mutual agreement. This relation is not recursive; i.e. the underneath servers can represent different taxonomies in each institute. This relation will be used to maintain a hybrid namespace (names have different aliases given by different institutions) when navigating the whole model searching for resources of specific topics. "Common Servers" should not have "Same" relations with each

other. The common part should not bear any kind of hybridity.

Figure 1 illustrates the hierarchy of "Common Servers" and "Institutional Servers" and the "Same" relation.

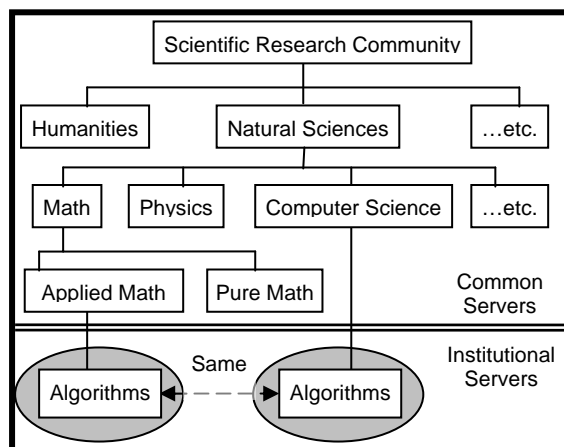


Figure 1. "Common" and "Institutional" Servers and "Same" relation.

The "Same" relation is assumed to be transitive; so it is the responsibility of all the currently "Same" institutional servers to approve a "Same" relation between one of them and a new server. This could be simple; one of those servers that are linked together through a "Same" relation nominates a new comer and announces it to the rest.

The new "Same" relation should be approved by all servers in the "Same" group (this rule is maintained by administrators). This will guarantee that all "Same" servers are really the same in their perspective towards the common area of interest.

"Same" relations between servers will be navigated from the underneath servers to find candidate "Same" servers in other parts of the hierarchy (same topics classified differently by different institutions). Explorative tools can help new servers nominate candidate "Same" servers.

Administrators, then, decide to apply for "Same" relations with some nominated servers according to their profiles (brief narrative expression of their interests). Figure 2 illustrates the utilization of "Same" relations between parent servers.

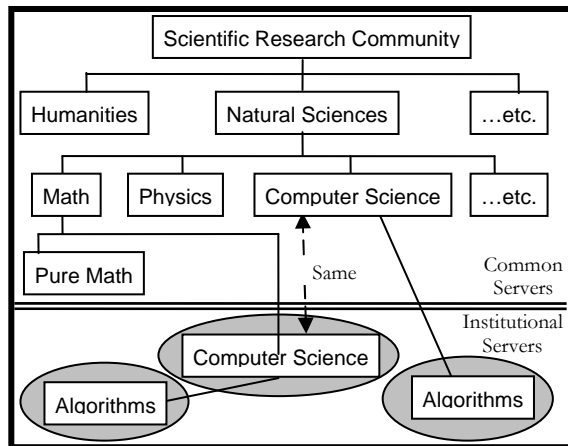


Figure 2. Utilizing "Same" relations between parent servers.

B. Introducing Data Grids to the Model

The groups of "Same" servers can establish data grids to share their resources. These joint data grids will be built away from the model. The only opportunity for those data grids to interact with the model is throughout a standardized programming interface.

A data grid initiated by same servers (or an already existing one) could be registered at any server. The registry holds five fields:

1. A unique identifier (under the server where it is registered). This server will act as the register of this data grid. The data grid full name is the data grid name prefixed with the hierarchical name of that server.
2. URL of the server that manages that data grid.
3. A profile to express the data grid mission and interests.
4. URL of the proxy agent code that represents the data grid in the model.
5. Ontology field to express the data grid semantics.

Traditional data grids that are built by "Institutional Servers" linked with "Same" relations will maintain their local resource namespaces to guarantee uniqueness of their resources' names. Externally, those resources are known by their full names. Resource full name consists of the resource local name (given by the traditional data grid it belongs to) prefixed with the data grid full name. Therefore, this resource full name is guaranteed to be unique in the whole hierarchy.

Partially equivalent research areas and topics in different disciplines can be linked through a "Similar" relation. "Similar" relation is neither recursive nor transitive. Searches that are not limited to a specific discipline can follow this kind of relations. (Search parameters and engine configuration can govern this behavior.)

The Grid concept is introduced to the model through the three kinds of relations: "Child", "Same" and "Similar". The "Child" relation maintains a simple hierarchical namespace (taxonomy) and the two semantic relations ("Same" and "Similar") maintain gray-scaled hybridization to the namespace. "Same" relation expresses the aliasing component of the hybridity while "Similar" relation expresses the partial commonalities between different branches. First order "Similar" server has higher possibility of having common interests and

contents than a second order one ("Similar" of "Similar"), and so on.

C. Semantic Integration

Semantics are expressed in the model in two ways:

1 Semantic Infrastructure:

The semantic schema built by the "Same" and "Similar" relations is the infrastructure for all search operations.

2 Semantic Interface:

The field "Ontology" which describes the server's mission and the data grids and resources' content and interests is added to allow agents (not part of the model) to navigate the model and evaluate its data. This is a general purpose field and its use is not restricted by any attempt of unification. It will follow whatever "Ontology" field policies known by its time; the admins could use whatever "Ontology" standards or *fashion* known to attract more agents to shop around the model. If there are many standards, many values could be added to the field (LDAP feature) tagging each of them by the name of the standard it follows.

Also, different groups of servers in the model could follow different "Ontology" standards. For instance, servers at USA could adopt a different standard than that adopted by servers in France. Or, "Ontology" of resources and data grids could follow different standards than "Ontology" of servers; they describe different domains.

Semantics will increase interoperability between the model and other models' agents searching the web for information. The field "Ontology" will be exposed for agents that navigate the Internet to shop at their liking. This is a point of contact where third party application developers should be aware of the internals of the model. They will implement agents that login the model and fetch the "Ontology" fields of the servers, data grids and resources.

Although some methodologies to build ontologies acknowledge the need for an integration step or the importance of integration activities in the process of building an ontology, the important problems of integration remain more or less unsolved [46].

The three meanings associated to the word "Integration" should actually be de-fined using the following words:

1. Integration - In the case of building a new ontology reusing (by composing) other available ontologies.
2. Merge - In the case of building an ontology unifying knowledge of several ontologies into a single one.
3. Use/Application - In the case of integrating ontologies in applications. [46]

Nobody has yet solved the problems of mapping and importing ontologies well enough. And likewise so far nobody has succeeded in making the uber-ontology and convincing everyone else to use it. The dream of the Semantic Web vision is that someday there will be thousands or millions of ontologies around the web, and millions of instances of them. And these will all somehow be integrated automatically, or at least if they

aren't integrated on the semantic level, then there will be magic software that embodies that integration. In any case, the hope is that someday intelligent agents will be able to freely and seamlessly roam around harvesting this data, squishing it together into knowledgebases, and reasoning across them. But neither harvesting, nor squishing, nor reasoning can really take place without some level of semantic integration of the underlying ontologies. The most critical missing piece of the semantic web puzzle is a good tool – a good methodology -- for mapping between ontologies. There are papers on automatic ontology mapping, but these capabilities haven't made into the ontology design tools. This needs to happen. Until the process of integrating ontologies is less work than simply reinventing the wheel, we are not going to see much semantic integration on the semantic web. In short the vision of the semantic web as a decentralized fabric in which multiple ontologies interoperate, really hangs on a good solution to this issue. [47]

D. Dynamic Configuration

Dynamic configuration can be done to optimize the search cost. For example, if the administrator of a "Common Server" noticed that many of its direct "Institutional Server" children are linked with "Same" relations (representing a specific branch of science) then he can take the decision of creating a direct child "Common Server" for that branch of science then inform all the "Same" "Institutional Servers" in order to follow the new taxonomy. The "Same" "Institutional Servers" should inform their children to apply for child relations with the new "Common Server". After moving all its children to the new "Common Server", the "Institutional Server" should resign from the model and disappear. Figure 3 demonstrates deepening of the common part of the hierarchy.

This mechanism will lead to incremental deepening of the common (unified) part of the name space. The ease of use and the almost no cost to join will encourage institutions to voluntarily move toward a unified taxonomy.

To maintain a safe, easy and fruitful deepening, simple assumptions should be made:

- Non-leaf "Institutional Servers" cannot bear content. (This means they cannot join data grids as well). Otherwise, administrators of the "Institutional Servers" should handle the issue of their content after leaving the model. E.g., they could arrange with the administrator of the newly created "Common Server" to transfer their content to it. In this case, alias collisions should be resolved arbitrarily.

- Leaf "Institutional Servers" should not be considered when deepening the common part of the server hierarchy.

- The newly created "Common Server" should apply to inherit all the "Same" and "Similar" relations of the "Institutional Servers" replaced by it. (Some "Same" relations will remain after the deepening because some "Same" servers either were not children of the same "Common Server" or have chosen not to join the restructure).

- "Institutional Servers" that have "Same" relations with "Common Servers" should not be considered when deepening the common part. The new "Common Server" will not inherit "Same" relations with other "Common Servers". The common part should not bear hybridity.

All institutional servers that will be replaced should announce to all their contacts (Same and Similar server) to break their links with it and apply for new relations with the newly created "Common Server" (housekeeping procedure that is done when a server leaves the model.).

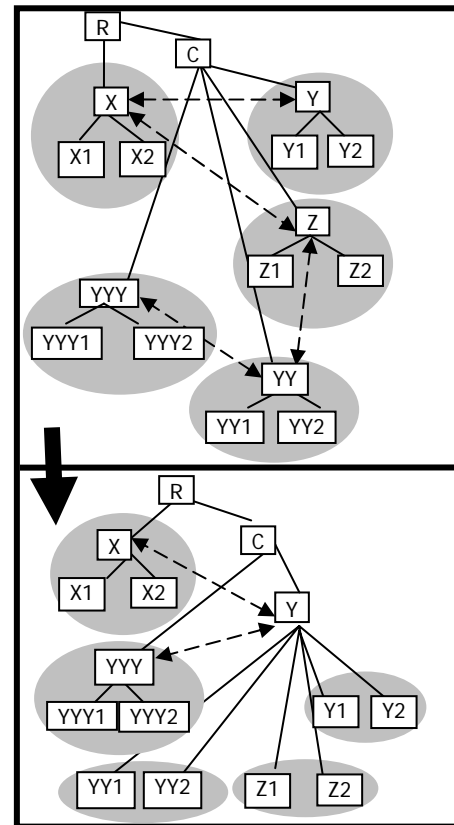


Figure 3. A "Common Server" is added to replace "Same" "Institutional Servers".

E. Wider Scope Collaboration

The data grids that are established by "Same" "Institutional Servers" can collaborate, as whole data grids, in a bigger collaboration model. They will benefit from the common name that will prefix all of their resources' names. Requests in the wider collaborative model will use the full resource names. Requests in small data grids will use the logical names without any prefix (they are traditional data grids). Data grids' internal business is completely transparent to the model. That's to say, the model will provide the data grids with an enabling for collaboration with each other and with standalone servers in the model. This enabling is a well-defined interface for a proxy agent that will act as a user to query resources on servers and data grids; and to respond to queries made by users and/or other data grids proxy agents.

The larger collaborative model can contain many data grids. "Similar" relations between "Institutional Servers"

in different data grids can be looked at as intentions for wider collaboration between them. Data grids can establish the wider collaborative model easily since the resource full names are already unique. Data grids can cache replicas of each other (for read only). They can serve their local users by providing those external resources. Proxy agents can reside into the data grid servers to represent the other data grids.

The information services and resource brokerage logic of each data grid will be encapsulated into the proxy agent it exposes. Actually, the proxy agent, as defined by this model, is a unified interface of inter-data grid gateways.

F. Usability of the Model

Explorative tools can be designed to help users and administrators navigate the huge hierarchy of "Common" and "Institutional" servers. Querying server resources is a kind of informed search. The user queries a resource alias (unqualified name) and the model will search for it on the local server and through the "Same" and "Similar" relations. Search results are sorted according to relevance. The ones found on the local server come first, then those on "Same" servers, then the ones found on "Similar" servers. The resources found on "Similar" servers are sorted according to the number of hubs ("Similar" relations) jumped from the local server to reach that "Similar" server. Since the "Similar" relation is not transitive, then, the direct "Similar" is more similar than the "Similar" of "Similar" and so on.

Sidekicks could be used to look for the same resource on "Similar" servers of each "Same" server that was found to have the resource and on "Same" servers of each "Similar" server that was found to have the resource. This mechanism utilizes the decisions made by other server administrators to better inform the informed search.

A user hooked to an "Institutional Server" can login as a user of the small data grid if there is one built by his local server and the other "Same" servers. In such case, he will be using non-prefixed logical resource names. This login will be transparent to the model and completely managed by the data grid facilities. He can also login as a user of a wider collaborative model in which his local data grid is collaborating with other data grids. He will be using prefixed resource names in this case to differentiate between resources from different servers and data grids. He can also login to surf the whole hierarchy of servers using the explorative tools to discover institutions, data grids and wide collaborations. This surfing will help server administrators make decisions on joining data grids, creating/deleting "Same" and "Similar" relations with other servers.

Resource querying and discovery through data grids is isolated from the normal server user resource querying and discovery. This is to allow data grids to use their "grid-power" facilitating the searches and retrieval performance. As data grids, they can export their powerful features (like replication and indices) to each other. They can let their agents (which are logged as normal server users) navigate the server hierarchy to shop

for resources acquisition. Acquired resources will be served for data grid users with all the power of data grids.

G. Security of the Model

Data grids inside the model are responsible of their security. They need to implement their own security models on their servers and on the stubs that handle their exported proxy agents as well. The model depends on the directory server security system. The model's security adopts recursive trust scheme for scalability. Each server authenticates its child servers (through certification and authorization). It brokers its authenticity at its parent's side when broadcasting their requests. It also brokers its authenticity at its child servers when multicasting its parent's requests to its children. A server, then, inherits its authenticity from its parent server while imposing its authenticity on its child servers.

This decentralized authentication pattern guarantees no bottlenecks known in the centralized security models. It puts no constraints on the model's growth and allows easy scalability. The cost for that is the multiple authentications needed to communicate with servers at a significantly far breadth distances across the hierarchy. This cost is limited by the depth of the hierarchy ($O \log N$) so, will always be affordable. Also, a caching technique could be used to cache credentials for future use.

H. The design goals of the Model

The major design goal is to make the collaborative model as much attractive for servers from different institutions to join as possible. The groups of contributors that are currently in data grids will be able to join and keep their data grid as is. Other groups that are currently unable to establish data grids will be able to establish relaxed collaborations through "Similar" relations. They can also negotiate "Same" relations with each other so that they can establish data grids.

Another design goal is hybridity; many fields should be able to benefit from this model like transportation and movie productions. Products of both industries are classified differently around the world. A legislative authority in a given country needs to set operational standards and tax bracketing for different transportations products. Also, a marketing company needs structured reports on different markets' potentials as per different "categories" of transportations products. Without a global collaborative community that allows all manufacturers to expose their data according to their preferred classifications while automating and facilitating the search effort, search could cost much while stay non-comprehensive.

Any field should be entitled for benefiting from this model if there are producers that do not follow a unified classification in a way that makes the global view unstructured enough to be handled by a normal collaborative model like traditional data grids.

Extensibility is another design goal. New semantic relations (other than "Same" and "Similar") can be suggested and added to the model to extend the semantic

of the inter-server relations such that more opportunities of collaborations can be exploited.

Decentralization is another design goal. This guarantees scalability of the model. The fact that the schema is decentralized and no central place holds complete information about the structure of the whole model makes every server responsible only of a limited number of nodes. (Nothing similar to Replica Location Index RLI tables is included in the proposed model). This eliminates the possibility of bottle necks. Also, "Common" servers at the root of the hierarchy are read only, so, simple mirroring solutions can scale up their affordability without the need for complex data grid solutions like distributed hash tables (DHT).

III. DESIGN

Server design is unified for all levels and roles (Root/Intermediate/Leaf, "Common"/"Institutional"). Certain functionalities are specific for certain roles.

A. Server Architectural Design

Figure 4 shows high-level server architecture. Proxy Agent Module is a framework for data grids to export their data content to each other through this collaborative model and to search for resources around the model on behalf of their data grid users. Data Grid application developers need to implement the Proxy Agent Interface. This is a point of contact where applications and their developers need to get awareness of the collaborative model.

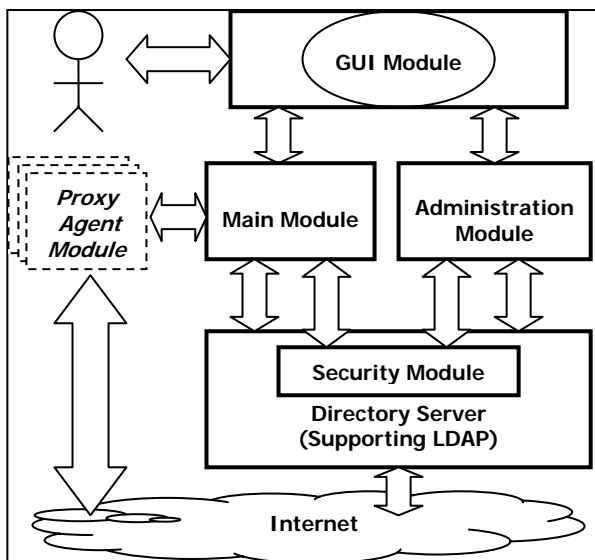


Figure 4. High-level server architecture.

B. Major Server Components

1 Explore Server

It is one of the exploratory tools started by the administrator. It sends a message to the local server telling it to explore the servers that have specific keywords in their profiles. The local server will create the exploration request and returns a key (a unique identifier) to the caller for future use (to browse the results as it is

being collected and accumulated). The local server (merely the Main module) will propagate the exploration request among the connected servers (the parent and the children) with the key attached to it. The local server will collect the exploration results from the connected servers as they send it. The connected servers will propagate the exploration request among their own connected servers (the parent and children except for the server from which it received the request). They will reply with a found message if their profiles contain matched keywords and will backward the replies to the server that has originated the request. A timeout (adjustable by the Administrator) will be used to control the collection of the offline replies. When user releases the key, the local server purges the attached results. Figure 5 illustrates the Sequence Diagram for the Explore Server tool.

2 Data Grid Proxy Agent

An interface exporting the following functions:

- QueryResource(Name): Queries the data grid for the resource name. If found, it returns resource information. This serves the Servers' users and other Proxy Agents.
- ReadResource(Res): Retrieves the resource.

3 Explore Data Grids

Finds all data grids by keywords in their profiles.

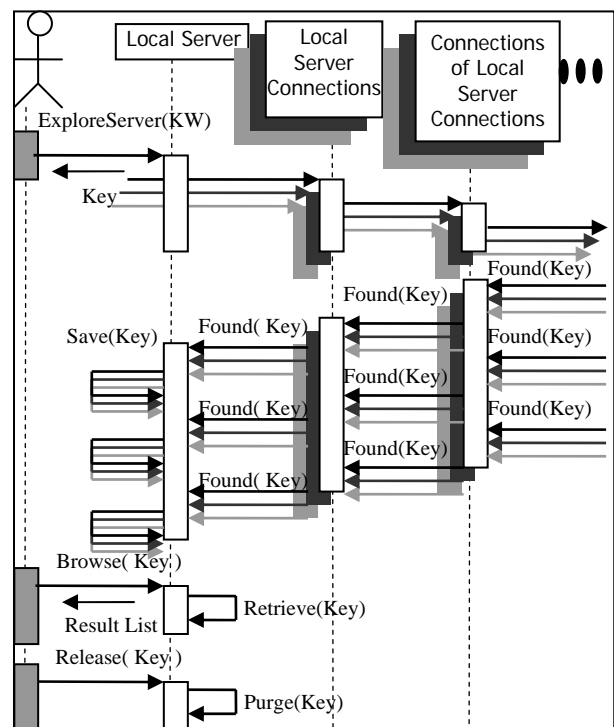


Figure 5. Sequence Diagram of the Exploratory Tool.

D. Use Scenarios

1 Installation (Administrator)

While installation, the Administrator will define the server type ("Common" or "Institutional") and set the Name and Profile data items. Then, Administrator will request child relationship with a server with a given URL. Once the parent approves the request and sends the new

ID, the server is hooked to the model. Figure 6 illustrates this scenario.

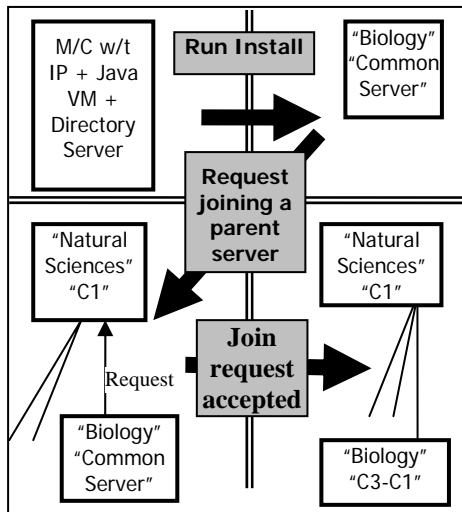


Figure 6. Installation scenario (Admin).

2 Setup (Administrator)

The Administrator will run the exploratory tools (in Administration Module) to find out candidate “Same” and “Similar” servers and data grids all over the model. The exploratory tools will let the Administrator browse the hierarchy throughout the hierarchical relations (Child/Parent) and the semantic relations (“Same” and “Similar”). Administrator will decide to request “Same” and/or “Similar” relations with other servers and will negotiate joining existing data grids (with DG managers). Later, the Administrator could get involved with other “Same” servers’ administrators in constructing new data grids. Administrator of common servers can make decision on deepening the common part of the hierarchy from their position if the number of “Same” direct child servers reaches a certain threshold. The detection of this case will be automatic by monitoring the children in the background. Once the threshold is reached, the Administrator will be consulted to make the decision. This scenario should be repeated periodically to get new updated results as the model will dynamically grow and change (new servers may join and new data grids may be registered).

3 Resource Discovery

Users will search for resources all over the model. They can join as users of data grid. In this case, they will be served by their data grid (which is considered a user on the model). Their data grid will provide them with its local resources plus the resources on servers and other data grids registered on the model. The data grid will look around through its proxy agent to find resources on servers and other data grids and collect resources for their users. In this case, the users are served indirectly by the model although they are not aware of it. Figure 7 illustrates the data grid resource discovery.

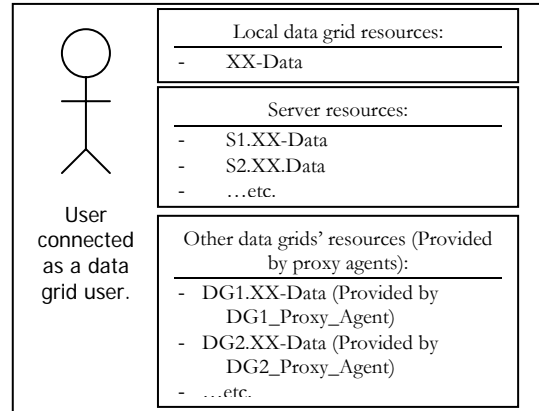


Figure 7. Data grid resource discovery.

Or, they can join as traditional users of the server. In this case they can issue queries that will follow the “Same” and “Similar” relations to bring information about resources. (Of varying degrees of relevance – resources on “Same” servers will be more relevant than resources on “Similar” servers). The resources provided by the direct “Similar” servers (1st order “Similar”) that have “Similar” relations with the user’s server are more relevant than the resources from the servers that are “Similar” to “Similar” servers (2nd order “Similar”) and so on. At each server reached through “Same” or “Similar” relations, the query could branch for a sidekick sub-query which looks up “Similar” servers of that server in case it was reached through a “Same” relation or “Same” servers otherwise. At each reached server, the query (or sub-query) searches for resources stored by the server or by data grids registered at the server. The full name of the discovered resources reflects their origin. Figure 8 illustrates server resource discovery.

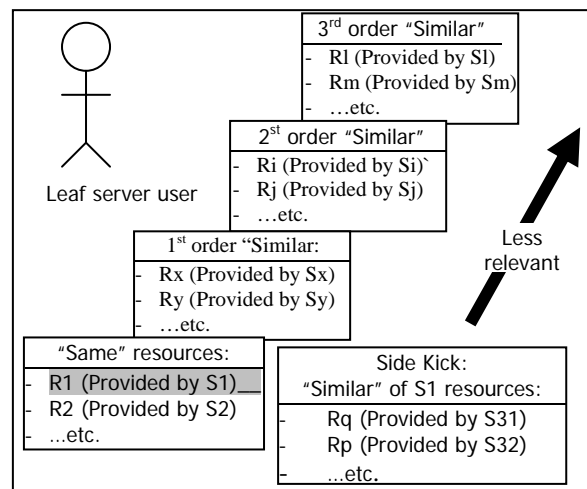


Figure 8. Server resource discovery.

In both scenarios, users will always be able to access local resources and remote resources. Remote resources accessed from the other model servers are prefixed by their host server full names and are profiled as per those servers’ profiles (the profile statements that are published by those servers). The resources accessed from data grids are prefixed with the data grids that provide them and are

received message with assumption to accept whatever the sender server proposes (request to be a “Child”, “Same” or “Similar” server). Resource discovery instructions are simulated before and after instructions to show the effect of the new relations on the discovery of resources. Also, Pausing instructions are used before and after restructuring instructions (deepening, joining, leaving) to let the user take snap shots of the hierarchy using the GUI. The simulator logs the status and results of every instruction.

V. SAMPLE OUTPUT

Figure 11 shows the hierarchy of servers representing scientific fields.

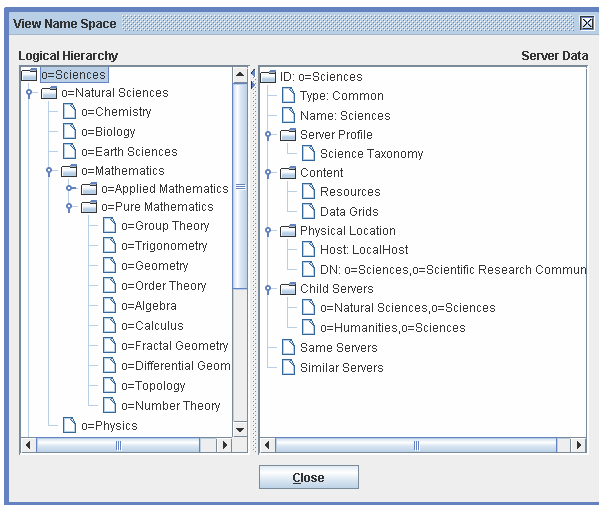


Figure 11. Sample Output.

Figures 12-14 show the deepening of the Common part explained earlier in Figure 3.

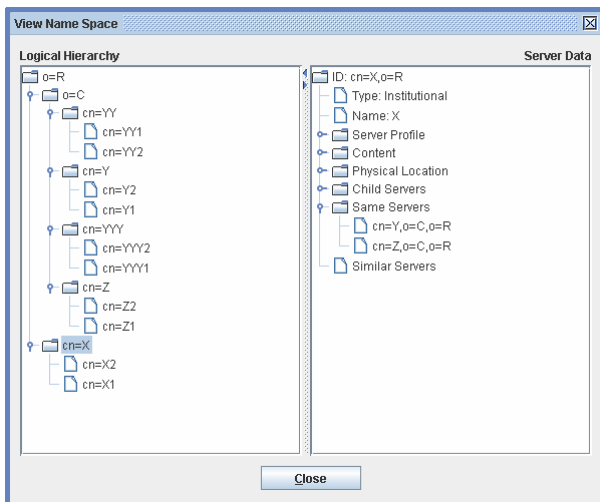


Figure 12. Before Deepening (X is “Same” to Y & Z).

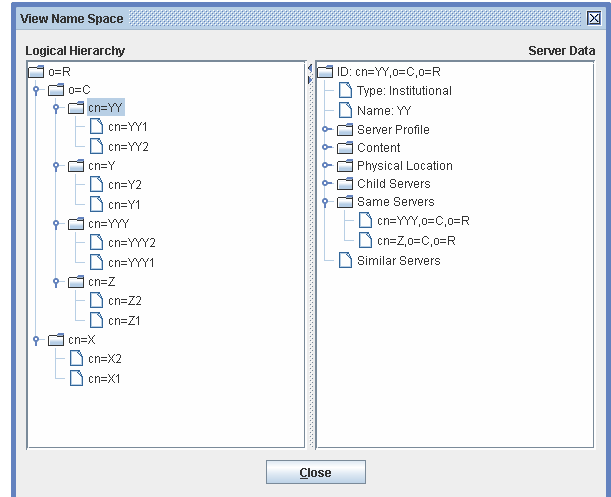


Figure 13. Before Deepening (YY is “Same” to YYY & Z).

Server Y becomes “Common” and parent of all YY and Z child servers. Z and YY will then disappear.

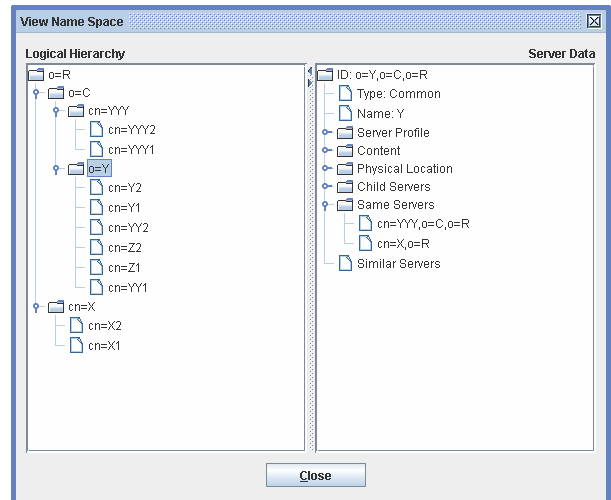


Figure 14. After Deepening (Y is Common, Z & YY disappeared).

Figure 15 shows a screen shot of the server resource discovery with sidekicks (illustrated in Figure 8). Discovered resources are sorted by relevance.

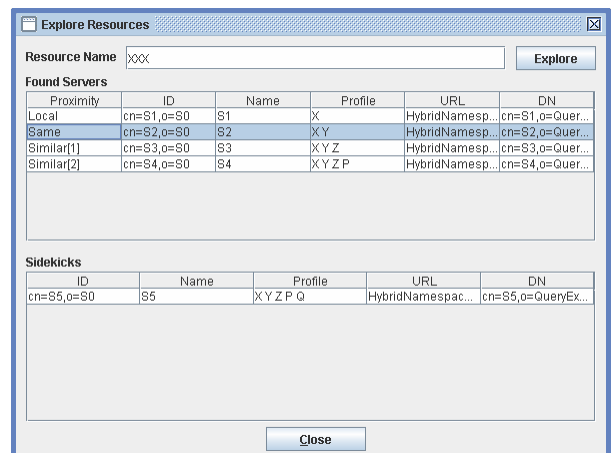


Figure 15. Server Resource Discovery.

VI. RELATED WORK

Few Data Grid projects focus on fusing heterogeneous Namespaces and solving the above taxonomy problem. The following surveyed set of previous similar works can be classified into five major categories: Semantic Grids, Knowledge Grids, Service Grids, Grid File Systems, and Agent-based solutions.

Semantic Grids are extension of the current Grid in which information and services are given well-defined meaning, better enabling computers and people to work in cooperation. This makes it easier for resources to be discovered and joined up automatically, which helps bring resources together to create virtual organizations. The descriptions constitute metadata and are typically represented using the technologies of the Semantic Web, such as the Resource Description Framework (RDF). Examples of Semantic Grid projects are: myGrid [9], Combe Chem [15], MAPGrid [17], Meta-data tools [18], Ontogrid [19], Dart Grid [20], the Ontology-based Semantic Resources [21], and the P2P Semantic Link Network [22]. This work is intended with semantics in mind; all items in the model (resources, servers, and data grids) are viewed in two ways: the regular keyword description (via a Profile field) as well as semantic description (via an Ontology field). In this research, the resources namespace consists of the resource names and the host names (servers & data grids). Therefore, ontology should describe each item. Nonetheless, the hybridity of the namespace is tackled through a semantic schema that expresses degrees of similarity between resource hosts.

Knowledge Grids are models for sharing and managing globally distributed knowledge resources. These models organize knowledge in several spaces, and provide knowledge grid operation languages for creating knowledge grids, putting knowledge to them, editing knowledge, to partially or wholly open their grids to all or some particular grids, and to get the required knowledge from the open knowledge of all the knowledge grids. These models enable people to conveniently share knowledge with each other when they work on the Internet. Examples of Knowledge Grid projects are the XML-based Integration Protocols [23], the XML-based Wrapper Mediator Integration Model [24], the XML-based Metadata Knowledge Grid [25], the Concept Data Objects Knowledge based Grids [26], the XML-based Integration tool [27], and the LDAP-based Integration Model [28]. This work starts with creating a static semantic schema using a fixed set of relations that represents the grayscale of the degree of similarity ("Same" and "Similar" as suggested herein). However, dynamic semantic schema is planned in the future (see Section VIII- Future Horizons). User defined relations should be able to introduced to represent further aspects of relations between resource hosts. This is a way of sharing knowledge about resources and their hosts.

Service Grids are XML integration formalism exposed as Grid services within the architecture. They are service-based architectures for providing data integration in Grids using a decentralized approach. They are based on the

well-known Open Service Architecture (OGSA) that aims to define a core set of capabilities and behaviors. Examples of Service Grid projects are Namespace service [29], Combining the Notification and GridFTP services [30], GridFS [31], the Metadata catalog Service Grid Registry [32], the Storage Grid on-demand interoperability [33], the Grid Integration Model [34], and the PPerfGrid Service [35]. This work is intended to serve data manipulation not service (or computing) facilities manipulation. Therefore, the only concern is about the services that manages the model. In Section VIII (Future Horizons), a provision of web services interface is suggested for managing the model.

Grid File Systems are name services that enable construction of a uniform, global, hierarchical namespace, a key feature needed to create a file-system grid. Combined with other grid replication and location-lookup mechanisms, it supports independence of position for users and applications as well as transparency of data location in a scalable and secure fashion. This name service enables federation of individual files as well as file-system trees that are exported by a variety of distributed file systems and is extensible to include non-file-system data such as databases or live data feeds. Such a federated namespace for files can be rendered by network file servers, such as NFS (Network File System) or CIFS (Common Internet File System) servers, proxies supporting the NAS (network-attached storage) protocol, or grid data service interfaces. Examples of Grid File System projects are the GForm Grid File System [36], the SDB Resource Model [37], the PVFS2 [38], the X-SIGMA [39], the GDIA [40], the VIG Schema mapping [41], and the Virtual Access and Integration tool [42]. This work handles hybridity of namespaces that is not tackled by Grid File Systems.

Agent-based solutions are multi-agent systems that provide service for publishing and discovery data in grids through metadata. These services are being developed in context of a grid middleware that explores the mobile agent paradigm as a way to overcome the design and implementation challenges of constructing a heterogeneous grid middleware. Examples of Agent-based solutions are MagCat [43], the SRB Storage Broker [44], and My SRB Storage Broker [45]. In this work, the data grids are introduced to the model through a simple agent interface that is responsible of exporting their resources to other servers and data grid users as well as shopping for resources at other servers and data grids.

VII. CONCLUSION

From the experimental results, the model is proven to work with almost no cost. Contributors have nothing to lose (and everything to gain) by joining the model. They need not pay any effort to adapt to the model; they join as they are, even those contributors with extremely odd taxonomies can still join, and exchange resources. A by-product of the design goals (decentralization) is the perfect performance measures. Since a server knows nothing except its content and the coordinates of its

neighbours (parent, children, "Same" and "Similar" servers), the operations are done faster.

The heavy search operations (resource queries and server explorations) are done by many servers through propagation patterns that follow child/parent relations and/or "Same"/"Similar" relations. This propagation involves more servers in the search operations exponentially. For example, at time t_1 neighbours of the first server will be involved (a matter of 10 servers). At time t_2 , neighbours of those neighbours will be involved (a matter of 10×10 servers) and so on. This exponential invocation reduces the search time to logarithmic cost ($\log n$) instead of linear cost (n , where n is number of servers). This guarantees no bottlenecks and fast searches. Moreover, the semantic infrastructure expressed by the semantic schema that is built by the semantic relations "Same" and "Similar" side by side with the semantic interface expressed as the field "Ontology" describing the server's mission, the resource content and the data grid interests, puts the model in the fourth quadrant on the scale of interoperability versus the scale of data and computation, and makes it comparable to the Semantic Grid in that sense. Knowledge and semantics allow better resource discovery by automating it, hence results in easier and faster resource discovery, which in turns results in higher interoperability. This increases interoperability between the model and other models' agents searching the web for information. The aliasing problem is solved by the semantic relation "Same" while "Similar" relation solves the issue of the partial commonalities between different branches.

VIII. FUTURE HORIZONS

A new operation "Fork", to divide a server into two or more servers, could be done in the future. This should help administrators cope with the dynamic change of the field description. Housekeeping needed after this operation is dividing the content (resources and data grids) and handling the relations. Dividing the server content among the new created servers is easy done by administrator. Handling relations needs some automation because it involves other servers. Announcements should be sent to parent, children, "Same" and "Similar" servers beforehand. Some "Similar" relations could upgrade to "Same" in case that a server was "Similar" to the forked server due to partial sameness of interests. After the "Fork" operation the server could find that it has same interests with one of the new created servers. Also, a mobile agent could be designed in the future to move around the model and collect information that would facilitate the future search of a specific server. This agent can be augmented by some provisions about the future needs of its owner server that guides its tours towards more fruitful navigations. Provisions can be stated directly by server owners (human intelligence) or by heuristics built from analyzing users' requests (artificial intelligence).

The semantic relations (currently "Same" and "Similar" only) could be extended in the future. An extensible framework for inter-server relations that

allows new kinds of relations to be added in the future could be standardized. This will be a way of keeping and exchanging knowledge about resource hosts. XML could be used to define the relations in terms of a set of standard attributes. Instead of hard-coded behavior for "Same" and "Similar", the navigating query (or agent in future) can fetch the existing relations on the hosting server and discover what actions could be done with them by understanding their semantics (Ontology field will be needed to describe the relation). A possible new kind of relations is "Dependent". Education, for example, could be looked at as a "Dependent" of "Statistics". The new relations should be publicized in a central place (hosted or pointed to by the root) for distinction. Local relations could also be declared in any server (as an application extension to the server by third parties) to be applied only in the sub tree rooted by it. This will be another point of contact where applications and developers should get awareness of the model. The future provision of adding new relations by third parties will create new roles for the model as per other models' needs. A future model X could suggest adding a new relation Y in the model to be used by the agents of the model X doing special processing or search routing. For example, new relations could be suggested to carry weights to express probabilities or, contrarily, costs to express improbabilities to support heuristic search techniques using different evaluation functions. Another future extension to this work is to move the complete, extensible and open messaging unit in the current design to be a separate middleware layer or subsystem. In such case, it will be easier for third parties to join and collaborate using the model. A bunch of web services could be designed to provide services like *DefineRelation*, *PublishRelation...etc.*

Caching is another possibility for performance enhancement in the future. Credentials and physical locations of remote servers (other than parent and child servers) could be cached for future use instead of paying the effort of getting them every time. A semantics web service could be designed at each server to externalize all the ontology fields under this server (the server ontology field, ontology field for each data grid registered by this server, an ontology field for each resource hosted by this server and the Ontology field that will provide the semantics of the new relations defined on this server). The parameters are to state the ontology language and the target element (resource, data grid, relation or the server itself). This should encapsulate the implementation details of the ontology field and should follow the contemporary standards for seeking and consuming semantic data. Even more, several web services could be designed to provide the same thing but following different conventions.

REFERENCES

- [1] Foster, C. Kesselman, S. Tuecke. *The Anatomy of the Grid: Enabling Scalable Virtual Organizations*. International J. Supercomputer Applications, 15(3), 2001.

- [2] Special Session on *Collaboration Grids and Community Networks*, A Joint Call from CTS'06 and the Global Grid Forum. The 2006 International Symposium on Collaborative Technologies and Systems (CTS'06) May 14-17, 2006
- [3] *The CombeChem Project*. www.CombeChem.org.
- [4] *D-Grid Initiative*. www.d-grid.de
- [5] The EU *DataGrid Project*, <http://eu-datagrid.web.cern.ch/eu-datagrid>.
- [6] *EnterTheGrid*. www.enterthegrid.com
- [7] *GGF* www.ggf.org
- [8] *Grid Computing Info Centre*. www.gridcomputing.com
- [9] *myGrid*. www.mygrid.org.uk
- [10] *Semantic Grid Community Portal*. www.semanticgrid.org
- [11] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, S. Tuecke, *The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets*, the Journal of Network and Computer Applications, 2001.
- [12] Tony Hey, Contributing Editor, *GRIDtoday* Special Features. *Working Toward a Semantic Grid*, March 28, 2005
- [13] [EUDataGrid] The EU DataGrid Project, <http://eu-datagrid.web.cern.ch/eu-datagrid>
- [14] http://en.wikipedia.org/wiki/Data_grid
- [15] [CombeChem07] *The CombeChem Project*. www.CombeChem.org
- [16] [myGrid07] *myGrid* www.mygrid.org.uk
- [17] Yun Haung, et al, "Supporting Mobile Multimedia Applications in MAPGrid", ACM International Wireless Communications and Mobile Computing Conference, Honolulu, Hawaii, 2007
- [18] Alexander Woehrer, Peter Brezany, Herbert Schentz, Martin Bloechl, Andreas Langegger, and Wolfram Woess. *EcoJoiner- "a Grid-based data integration infrastructure for ecological domains. Functional specification for database access and integration architecture. Austrian Grid Deliverable AG-DM-4bA-6M-4a-2-2005 v1*, 2005.
- [19] Ontogrid www.ontogrid.org
- [20] Yuxin mao, Al et., "Semantic Browser: an Intelligent Client for Dart-Grid", Lecture Notes in Computer Science, May 2004
- [21] Tinger Chen, el at, "The Ontology-based Semantic Resources", Proceedings of the 19th IEEE International parallel and Distributed Processing Symposium, IPDP 2005
- [22] Hai Zhuge, Jie Liu, "A Novel heterogeneous Data Integration Approach for P2P Semantic Link Network" Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters WWW Alt. 04, May 2004
- [23] Jin Dejiang, "Integration Protocols for Heterogeneous Namespaces", Grid and Cooperative Computing, 2007. Sixth International Conference, V. 20, Issue 3, August 2007
- [24] Anastasios Gounoris, al et, "The XML-based Wrapper Mediator Integration Model", International Symposium on Applications and the Internet Workshops, (SAINTW'07), 2007
- [25] Carlo Mastoianni, "The XML-based Metadata Knowledge Grid", The International Conference on Hybrid Information Technology, V. 1, ICHIT'06, 2003
- [26] W. Moore, "The Concept Data Objects Knowledge based Grids", International Journal on Digital Libraries, Spriner Berlin, V. 5, n. 2, April 2001
- [27] Comito Carmela, Talia Domenico, "XML Data Integration in OGSA Grids", Lecture Notes in Computer Science, V. 3836 LNCS, Data management in Grids, DMG 2005
- [28] Gao Jinsong, Zhang Wen, Meng Lingkui, "A Method for heterogeneous Spatial Data Integration with Storage Agent in Grids", Proceedings of the 2005 International Conference on Wireless Communications, Networking and Mobile Computing, WCNM 2005, V.2, 2005
- [29] Omr Anderson, et al "Global Namespace for files", IBM Systems Journal, Armonk, Vol 43, Iss. 4, 2004
- [30] Hongwei Zeng. "Combining the Notification and GridFTP services ",International Journal on Digital Libraries, Spriner Berlin, V. 5, n. 2, April 2005
- [31] Marcelo Nery dos Santos, Renato Cerqueira, "GridFS: Targeting Data Sharing in Grid Environment", The Journal of CCGrid, Vol. 17, 2007
- [32] Peisheng Zhaoa, et al, "Conceptual Modeling and Metadata; Grid Metadata Catalog Service-Based OGC Web Registry Service", Proceedings of the 12th Annual ACM International Workshop on Geographic Information Systems GIS '04, November 2004
- [33] Yuhui Deng, Frank Wang, "Polemics: the role of transparency in distributed Services: Opportunities and Challenges of Storage Grid Enabled by Grid Service", ACM SIGOPS Operating Systems review, Volume 41, Issue 4, July 2007
- [34] Lin Weiwei, Qi Deyu, Li Yongqiu, "Grid-based Integration Model of Distributed and heterogeneous Data", Jisuanji Gongcheng Computer Engineering, V. 32, n. 24, December 2006
- [35] John Hoffman, Et Al, "PPerfGrid: A grid services-based tool for the Exchange of Heterogeneous Parallel Performance data", Proceedings of the 19th IEEE International parallel and Distributed Processing Symposium, IPDP 2005
- [36] Sato, Matsuoka, "Data Management on Grid File system for Data-Intensive Computing", International Symposium on Applications and the Internet Workshops, (SAINTW'07), 2007
- [37] Qingyang Wang, Al Et., "A New Architecture of Data Access Middleware under Grid environment", The IEEE Asia-Pacific Conference on Services Computing (APSCC'06), 2006
- [38] Chao Tung Yang, Al et., "Using Grid Computing and PVFS2 Technologies for Construction of an E-Learning Environment", Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies, 2005
- [39] Dongkwang Kim, Al et., "X-SIGMA: XML Based Simple Data Integration System for Gathering, Managing, and Accessing Scientific Experimental Data in Grid environments", Proceedings of the Second IEEE International Conference on e-Science and Grid Computing, 2006
- [40] Wei Xiaohui, et al, "GDIA: A Scalable Grid Infrastructure for Data Intensive Applications ", The International Conference on Hybrid Information Technology, V. 1, ICHIT'06, 2006
- [41] Qifeng Huang, et al "Building a Portable File System for Heterogeneous Clusters", Tsinghua Science & technology, Volume 10, Issue 1, Feb. 2005
- [42] Liu, Gui, al et., "A Data Access Scheme of Heterogeneous Data Resource in Grid", Grid and Cooperative Computing, 2007. Sixth International Conference, V. 20, Issue 3, August 2007
- [43] Bysmarck Barros de Sousa, et al., "MagCat" An Agent-based Metadata Service for Data Grids", Proceedings of

the Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID'06), June 2006

- [44] Stergios V. Anastasiadis, Syam Gadde, and Jeffrey Chase, "Scale and Performance in Semantic Storage Management of Data Grids", International Journal on Digital Libraries, Springer Berlin, V. 5, n. 2, April 2005
- [45] Arcot Rajasekar, Michael Wan, Reagan Moore, "MySRB & SRB Components of a Data Grid", Proceedings of the 11th IEEE International Symposium on High Performance Distributed Computing HPDC'02, July 2002
- [46] H. Sofia Pinto, Asunci3n G3mez-P3rez, Joao P. Martins. "Some Issues on Ontology Integration", Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID'06), June 2006
- [47] Nova Spivak, *The Ontology Integration Problem*, Minding the planet, www.mindingtheplanet.net, August 31, 2006

Dalia El-Mansy is currently a Ph.D. Student at The Southern Methodist University/Department of Computer Science, Texas, USA. She has M.Sc. From the American University of Cairo and a B.Sc. from the same University. Her research interests are in the areas of Parallel and Distributed Computing.

Ahmed Sameh is a Professor of Computer Science at The American University in Cairo. He holds a Ph.D. and M.Sc. from the University of Alberta Canada and a B.Sc. from the University of Alexandria. His research interests are in the areas of Parallel Processing, and High Performance Computing, Neural Networks, Mobile Computing, and Hardware Design.