

A Taxonomy of Data Management Models in Distributed and Grid Environments

Farrukh Nadeem

Department of Information Systems, Faculty of Computing and Information Technology
King Abdulaziz University, Jeddah, Saudi Arabia.
E-mail: fabdullatif@kau.edu.sa

Abstract—The distributed environments vary largely in their architectures, from tightly coupled cluster environment to loosely coupled Grid environment and completely uncoupled peer-to-peer environment, and thus differ in their working environments as well as performance. To meet the specific needs of these environments for data organization, replication, transfer, scheduling etc. the data management systems implement different data management models. In this paper, major data management tasks in distributed environments are identified and a taxonomy of the data management models in these environments is presented. The taxonomy is used to highlight the specific data management requirements of each environment and highlight the strengths and weakness of the implemented data management models. The taxonomy is followed by a survey of different distributed and Grid environments and the data management models they implement. The taxonomy and the survey results are used to identify the issues and challenges of data management for future exploration.

Index Terms—Data Management Tasks and Challenges, Data Management Models, Taxonomy of Data Management Models, Data Management in Distributed and Grid Environments.

I. INTRODUCTION

Distributed systems allow to harness the power of collections of autonomous compute and storage resources distributed over a network and connected through a middleware. The middleware enables sharing and coordination of the distributed resources by giving a perception of a single, integrated facility. The resources in a distributed system may observe different sharing and coordination policies. Under each distinct set of high-level sharing and coordination policies, a different distributed environment is formed, like cluster, Grid, peer-to-peer environment etc. With the growing maturity of electronic applications, e.g. e-Science and e-business applications, the development and utilization of such application is also increasing. Many of these applications produce and/or operate on a large amount of data, which is usually distributed over the network. Such data can be number, words, descriptions, measurements or only observations, etc. However, management of distributed

data in a given distributed environment is very critical for a its reliable and efficient access. Usually, the data is managed through a data management system that is responsible for systematic collection, organization, storage and access of the data. A data management system is also responsible for several other tasks required in different environments, including backup, recovery, replica management, etc. Some of the distributed environments are even especially developed for data management, like data Grids.

Depending upon the goals and constraints of a distributed environment, a set of methods is employed by the data management system to perform the data management tasks. This set of methods is referred to as a data management model. Thus, the role of data management model is very critical in overall management of the data in a distributed environment. Due to ever increasing size of data, the importance of choosing a right data management model is getting more than ever before. Choosing a right data management model can lead to an effective and efficient data management. Otherwise, an inappropriate data management model may lead to failure in obtaining the overall objectives of data management, like security, fast access etc.

To support choosing a right a data management model, in this paper, we present a comprehensive taxonomy of data management models in distributed environments. For better evaluation and comparison of the data management models, we first lay down the stage by describing major data management tasks and the data management system. Further, we focus on the core tasks in a data management model and describe the taxonomy of data management models in terms of these core tasks. The taxonomy highlights the data management requirements specific to each environment and highlights the strengths and weakness of the implemented data management models. To give an overview of current implementation of the described data management models, the taxonomy is followed by a survey of distributed and Grid environments and the data management models they implement. The taxonomy and the survey results are used to identify the issues and challenges of data management for future exploration. Last but not the least, we compare the presented data management models for their strengths and weaknesses.

The paper addresses all these issues in the following

sequence: section 2 describes data and major data management tasks in distributed environments, section 3 describes data management model in distributed environment. Section 4 describes taxonomy of data management models, section 5 presents data management models employed in existing distributed and Grids environments. The current challenges along with the future work are highlighted in section 6. Finally, we conclude in section 7.

II. DATA AND DATA MANAGEMENT

Before proceeding towards the taxonomy of data management (DM) models in distributed environment, to set the stage, we briefly describe data and its management aspects in the following sub-sections.

2.1 Data

Conventionally data is conceived as “a collection of facts from which conclusions may be drawn” [41] or “group(s) of information that represent the qualitative or quantitative attributes of a variable or set of variables” [40]. Such type of data is also called statistical data. In computer science, “data is anything in a form suitable for use with a computer. Data is often distinguished from an application source code which is a set of instructions that details a task for the computer to perform. In this sense, data is thus everything that is not application source code” [42]. In distributed and Grid environments data usually refers to the input and output files or streams. A data source refers to an entity generating data, for example, a scientific application, sensors etc. A data resource refers to a storage media holding the data, like disks, tapes etc. Different applications in the Grid need different types and size of data, such as, data intensive applications need large amounts of data while computation intensive applications usually utilize comparatively lesser amounts of data.

In distributed and Grid environments data exists in different types and nature such as input/output(I/O) files, access control data, version control data, security related data, and metadata. Figure 1 describes the taxonomy of data types in distributed environments. Input files contain necessary data that an application takes as input for its execution; output files contain data generated by an application. The output files generated by one application may further be treated as input file(s) for another application. Access control is the process through which the users are identified and granted certain privileges to information, programs, services, hardware resources, or systems. Access control data comprises of user names, user groups (e.g. Unix user groups), their access permissions and access logs. Version control data refers to data about changes to software, hardware, documents and other information stored as computer files, ensuring that a record of back-up files and historic copies are kept separately for backtracking.

Version control data aims at preserving the integrity and reusability of each version of data. It includes version number, time stamps and patches to each version.

Security related data includes user login data, proxies, user certificates, and public/private keys.

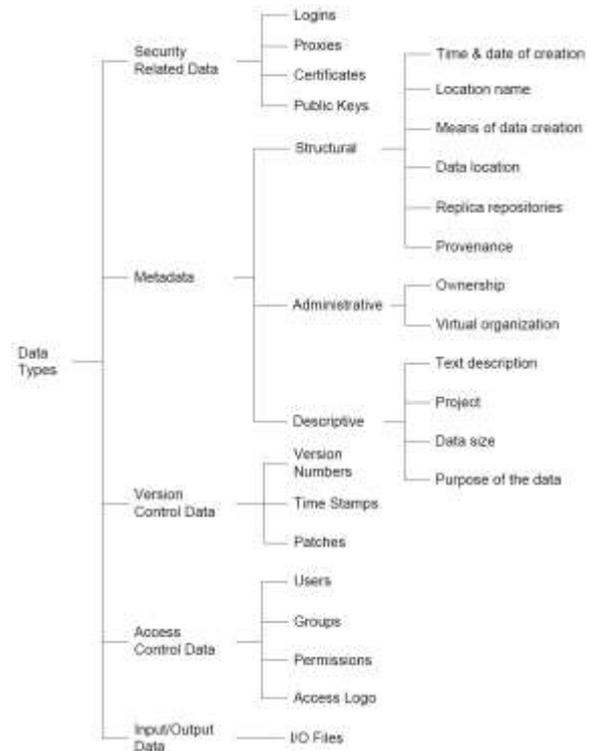


Fig.1. Taxonomy of types of data in distributed and Grid environments

Besides these types, *metadata* plays a major role in data management in distributed environments. The metadata refers to structured data that describes characteristics of the data. It is loosely defined as data about data. In distributed environments, one of the main objective of metadata is to enable fast access to the data. Metadata usually includes (but is not limited to) descriptive, administrative, and structural information about data, for example, description of data in plain text, data size, time and date of creation, ownership, means of creation of the data (like project and program name), purpose of the data, location of data and replica repositories, provenance etc.

2.2 Data management

Data management is a process of development, execution and supervision of plans, policies, programs and practices that collect, validate, organize, archive, control, protect, deliver and enhance the value of data and information assets [24]. A data management system is a set of computer programs that enable data management tasks (see section 2.2.1), maintain data integrity, and ensure timely access of the data. In an environment where compute resources and data resources both are distributed, the data management tasks are quite different from those performed in traditional data management systems. In distributed environments, besides other computations, the access to distributed data resources and their management are also treated as a vital functionality [21, 31]. The main functions of data management system in a distributed environment include:

- Provision of secure access to distributed data resources
- High performance transfer protocols for transferring data
- Mechanism for scalable replication
- Transaction processing
- Distribution of data on-demand [46]

Besides these functions, the system is also required to have: the ability to search through abundant available data sets, the ability to discover suitable resources hosting data sets and the ability to allow resource owners to grant permission to access their data resources [46].

2.2.1 Major Data Management Tasks

A *data management task* refers to a core functionality regarding data storage and retrieval, performed by a data management system. The main data management tasks in distributed and Grid environments are as follows:

- **Data Resource Discovery:** Data resources in large scale distributed environments like Grids may exhibit different properties like storage capacity, availability etc. On a data access request, it is important to discover the nearest available data resource that meets the given quality of service (QoS) and deadline requirements.
- **Distributed Query Processing:** The query processing deals with integrated data access and analysis for choosing an optimal data resource. The distributed query processing is of significant importance in a data intensive environment where data is stored at multiple nodes. In the traditional data management systems, the query processing deals with single data resources or a limited distributed database systems, but the large scale distributed environments like Grid, where there are multiple heterogeneous resources, require a special support for distributed query processing.
- **Data Security Management:** It covers the tasks related to data access management, data erasure, and data privacy. Data access management refers to managing rights for data access (authorization) and modification (integrity), and making data available “to the legitimate user in a controlled way” (authentication) when requested (no denial of service [43]). Rights for accessing data are commonly managed through access control mechanisms, like role based access control [27]. PKI [16] is widely used in the Grid to authenticate the users. Commonly used approaches to ensure data availability include backup, replication, etc. Data erasure refers to completely destroying (through software methods) all electronic data residing on a storage media. The approaches that ensure data availability also help in case of data erasure. Data privacy refers to legal and political issues related to data disclosure.
- **Distinguished File Names:** To distinguish between the files in a distributed environment like Grid, a special naming scheme is required, where no name

can be duplicated.

- **Data Sharing:** Not only the data, but the data resources also need to be shared. A data management system should offer shared access and monitor the issues related to the shared provision, like concurrency etc.
- **Data Consistency:** It refers to the accuracy, validity, and integrity of the data among users/applications and across the data resources and replicas. In a distributed environment any legitimate user may modify, move, remove, or rename any data at will. Data consistency ensures that each user observes a consistent view of the data, including the effects of his own transactions as well as any transaction by any other user/application.
- **Data Access Transparency:** It refers to a uniform access to the data irrespective of where it is located in the distributed environment and who created it.
- **Managing Context Attributes:** Such a management deals with context attributes so that the state information, generated by remote processes, can be managed. This is used to implement digital libraries and federate data Grids. One example of this management can be Storage Resource Broker (SRB). The SRB server maps the standard operations to the protocol of the specific storage system, besides a metadata catalog is maintained in any desired database technology [34].
- **Metadata Management:** Management of metadata corresponds to making sure that the metadata are current, complete, and correct at any given point in time.
- **Managing Digital Libraries:** Just like data repositories, the digital libraries also need to get managed and monitored by a data management system. As the data management in Grids is becoming increasingly important and for scientists and the users of Grids it is highly required to share large data collections, so digital libraries are maintained. Digital library is basically an archive of the large collection of data. A digital library maintains the digital entries under a collection. To access digital entries there should be some mechanism of authentication.
- **Schema Management:** Schema is an organization of data either on a network or in a database. Schema management includes the management of users, user groups, associative privileges for data access [37] etc.

2.2.2 Data Management Tasks for High Performance Environments

The following data management tasks are of particular interest for high performance environments.

- **Data Caching:** Data caching refers to placing data at a node that later offers an efficient access from the client’s node. Data is cached by analyzing the client’s access patterns and availability of the data as well as data resources.

- **Data Access Efficiency:** Data access efficiency refers to how quickly and persistently the data can be accessed. In the Grid, several methodologies are employed to optimize data access efficiency such as, replicating the data near the data access node, caching or streaming of data, pre-staging the data before the time the data is actually needed (e.g. during execution) and optimally selecting data resources.
- **Data Replication:** A data replica is a complete copy of the data that is relatively indistinguishable from the original. A replica contains all the properties of the data volume, including local shadow copy settings, security settings, and sharing. Data replication is commonly used to ensure high availability, reliability, fault tolerance, and efficient access of data.
- **Data Collection:** The data collection refers to more than one sets of data [40]. A data management system in large scale distributed environments like Grid, should provide the functionality of data collection for efficient referencing of the large data sets.
- **Fault Tolerance:** It is the ability of the data management system to respond gracefully to an unexpected failure of data resources. Data management system should be capable of taking backup, and recover from failure.
- **Load Balancing:** Workload (computational work done in performing DM tasks) balancing on different resources in the distributed environment is of critical importance for high performance and throughput. For load balancing, an efficient and scalable scheduling of DM tasks on available resources is required.

III. DATA MANAGEMENT MODEL

In a distributed/Grid environment, several DM tasks may be performed through different distinct approaches. The selection of these approaches is mainly governed by the objectives and constraints of the working environment. Besides, there are several other aspects to be decided (see Section 3.1) in the implementation of a comprehensive data management system. A data management model is an abstract representation of a set of distinct approaches to perform the DM tasks and implement a data management system. A data management model should not be confused with data models. A data management model describe the tasks related to the managerial affairs such as, networking, data organization and storage, data access etc. Whereas, the data models describe data elements and their relationships through different models like, entity relationship model, data flow model, flow charts model and semantic model etc. In this paper, we focus on data management models in distributed and Grid environments.

3.1 Major Components of Data Management Model

Figure 2 describes the taxonomy of DM model components. These components along with their different approaches and the design considerations for a DM model are discussed as follows. It should be noted that not all DM tasks are included in a DM model. It includes only those DM tasks that have distinct approaches to perform them and each of these approaches considerably affect the overall performance of the DM task.

- **Data Organization:** The data organization describes how data is physically stored on data resources. In a centralized organization the whole data is stored at a central place and is provided on demand through different mechanisms such as from a replica on the nearest data resource. In a distributed organization data is divided and stored at more than one locations, which are synchronized and made available using optimized discovery and allocation mechanism.
- **Span:** The span (or scope) of a DM model describes the number of administrative domains the DM model is covering. The single domain DM model has a common infrastructure that covers the entire domain and its span is referred as intra-domain. In the Grid environment there are multiple administrative domains, which usually exhibit different access policies and security related issues within each domain. In such an environment, the DM model is implemented as a generic infrastructure layer that provides a uniform interface to carry out DM tasks in different domains. The span of such DM model is referred to as inter-domain.
- **Metadata Management:** The metadata management is an important part of replica management. Beside the management view point, the usage of data also requires metadata as it is not humanly feasible to search the required data from the huge bulk available in a distributed environment. The logical file names are mapped to the physical files through the metadata available in replica. Based on the details stored, metadata may be classified as system defined or user defined. The system defined metadata consists of basic information about data usually generated through the underlying operating systems, like file type, file size etc. and is stored as a part of data. In user defined metadata, the users can set up self-defined metadata collection. This usually includes data properties, relationships and classifications, and is commonly stored as attribute-value unit triplets. Depending upon the “collecting agent” or “time of collection”, metadata collection is classified as active or passive. Metadata manipulated by the system is referred to as active metadata. It is updated immediately whenever there is any change/update in the data. Whereas, passive metadata is collected and updated only at user request.

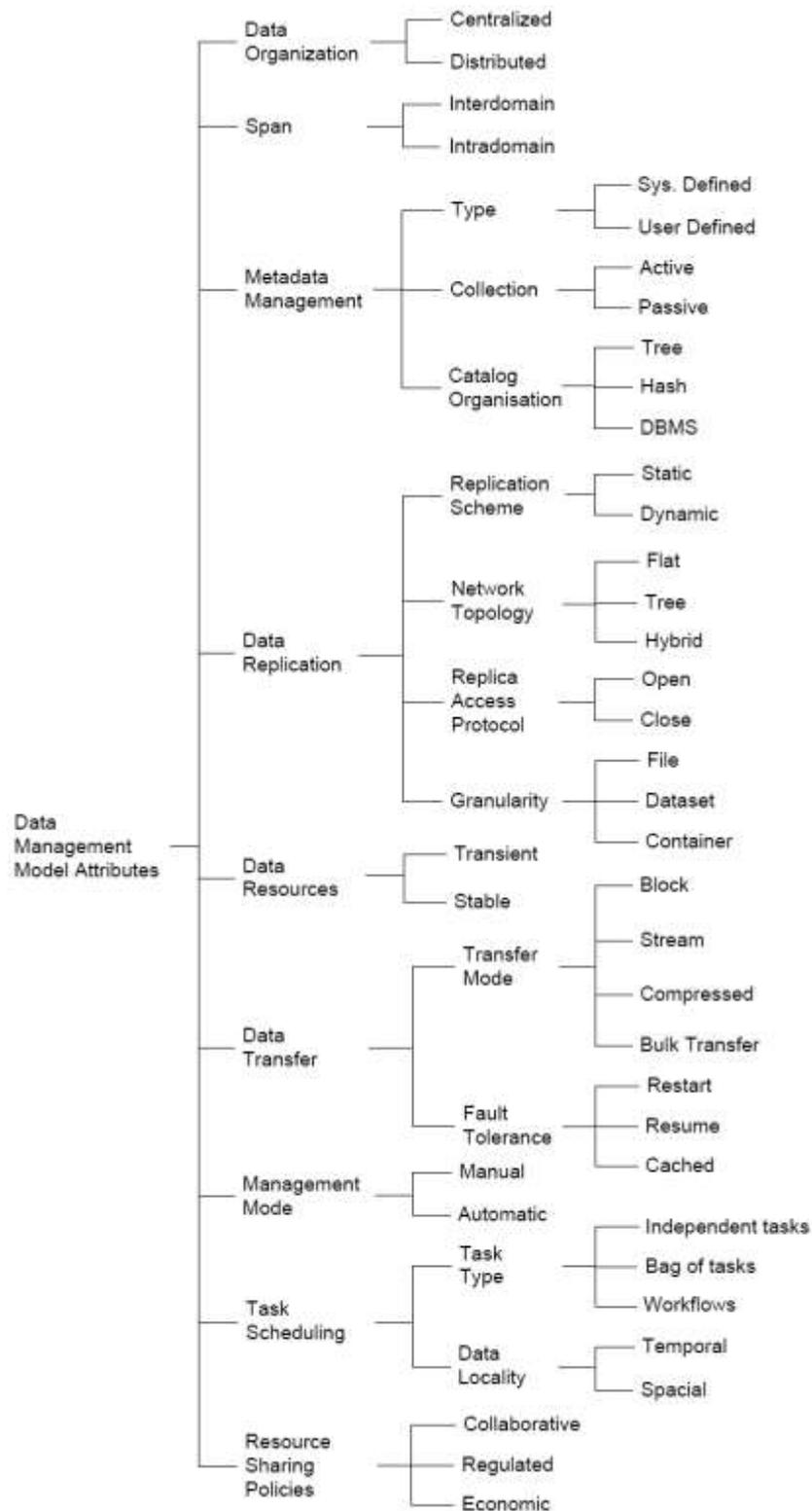


Fig.2. Taxonomy of DM model components in a distributed/Grid environment

- Data Replication:** The Data replication involves different data resources which are usually connected through high speed data transfer protocols. The major considerations for data replication management include metadata management, replication method, catalog organization, network topology, and replica access protocols, and are described below.
 - Replication Schemes:* There are two major schemes for data replication: static and dynamic. These two schemes exhibit a tradeoff between data availability and system performance. In dynamic replication

scheme, data is replicated as soon as there is any update. Where this scheme ensures high data availability, it occupies a significant network bandwidth and thus affects overall system performance. Therefore, application of this scheme is restricted to a small number of data objects – mostly the critical data objects. Dynamic replication scheme must be supported by sophisticated methods for recovery from faults, like network connection failure, and should be adaptable to changes in the network bandwidth requirements, data availability and storage accessibility. Under static data replication scheme, data is replicated on user request, which may be after fixed intervals or on demand. This scheme is employed in situations where data usage is not frequent and/or data is not critical. The main objectives of static replication scheme include improving the overall performance of the system by utilizing the network in off-peak hours. However, availability of the updates is compromised till the next replication.

- *Replication Catalog Organization*: The replication catalog is intended to keep track of replication events and manage the replication process efficiently. It includes data dictionary tables, tables of replicated events and other administrative information about replication. It may be organized as Tree, Hash, or DBMS style. In case when catalog access is provided through LDAP (Lightweight Directory Access Protocol) based implementation, it can be efficiently organized as a Tree. Grid toolkit Globus' [32] replica management organizes its replication catalog as a Tree [32]. Replicas can also be cataloged by using document hashes as in [47]. In a DBMS style catalog management system the catalog is stored as a part of a database.
- *Network Topology*: The replica containing data resources may be organized in variety of topologies, mainly flat, hierarchical and hybrid topology. Under the flat topology the data resources are connected like classical P2P fashion. The hierarchical topology arranges distributed data resources in a tree-like organization, where updates are propagated through specific data resources in the hierarchy. The hybrid topology combines both flat and hierarchical approaches.
- *Replica Access Protocol*: It lays a basis for further replica management tasks. The Open access protocol allows access and transformation of data independent of replica management system. The closed access protocol restricts the users to access replicas only through replica management systems and is mainly used in tightly coupled replicas.
- *Granularity*: The granularity of a replication system defines subdivision of data to address its management tasks efficiently. The replica management system that deals simultaneously with multiple files has a dataset level of granularity. The replication strategy may be defined to deal with one file at a time or even a fragment of file, which is a better choice for large files but requires additional techniques. In case replication system deals with a large number of small-size files, several files may be logically grouped into a container to achieve higher efficiency of replication.
- **Data resources**: A DM model is largely driven by type of the data resource. In case of transient data resources, the life time of data streams is very short. The data is transferred to the destinations only at a certain time of the day. In this case, the data management system employs such mechanisms that obtain the optimized benefits from this short lifespan of data streams. Special care is taken for execution of the applications accessing data from such resources. The stable data resources are intended for mass data storage, which remain available for 24 hours 7 days a week, like production databases. The setup for the management of such a voluptuous data resource is usually huge and complicated, and requires scalable methods.
- **Data Transfer**: One of the fundamental concerns in data management a distributed environment is transfer of data. It is not just the transfer of bits on the network, rather it is also about the secure transfer, transfer mode and fault tolerance.
 - *Transfer Mode*: There are four basic transfer modes: block, stream, compressed and bulk.
 - *Fault Tolerance*: In a distributed/Grid environment the fault tolerance mechanism is usually based on the type of data resources. The usual fault tolerance mechanisms are as follows.
 - ⊗ *Restart*: In this methodology the data transfer is restarted after recovery from fault. As a result the data already transferred is wasted and there are overheads of reconnection and retransfer.
 - ⊗ *Resume*: The resume methodology keeps track of data transformation in such a way that in case of some fault the transformation is resumed from a latest point before the fault. To achieve this functionality, different methods are available such as break points or last data bit information. For example, GridFTP [45] resumes from the last bit acknowledgement.
 - ⊗ *Cached*: In such a fault tolerance mechanism, “store and forward” protocol is implemented, in which the data to be transferred is cached first for a possible retransfer, and then transferred. For its caching of the data, this mechanism slows down the transfer and overall performance of the system.
- **Management Mode**: In large scale distributed environments like Grids, carrying out the data management tasks manually becomes very tedious and thus require to be managed in an automatic fashion. In autonomic management the monitoring and allocation of resources is automatically done

through predefined algorithms which are triggered on certain events. The management system needs no human data-administrator or network administrator rather the software acts as administrators. Such autonomic systems have got the property to take decision by themselves. They can also heal themselves when needed. However, many of DM tasks require more research to be performed automatically, and thus are managed in manual fashion. There are human administrators and managers to monitor and allocate the resources.

- **Task Scheduling:** The task scheduling refers to mapping of the requested tasks to the specific resources. The scheduling strategies in a DM model are usually classified on the basis of type of the task that is requested and locality of the data that will be accessed.
 - *Type of Task Request:* The type of DM requested task classifies the scheduling. There are three major types of tasks requests.
 - ⊗ *Independent Tasks:* It is the very first level of tasks in which there are individual tasks which need to be scheduled. Such tasks are the smallest independent computation units which are scheduled independent of each other.
 - ⊗ *Bag of Tasks:* In this type of request there are number of individual tasks (collectively referred to as bag of tasks) which are executed to achieve a single goal. All the tasks need to get scheduled in a way that the given constraints/targets, such as a deadline of the execution of bag of tasks etc., may be achieved.
 - ⊗ *Workflows:* In a workflow of tasks, the tasks have execution as well as data dependencies on other tasks and thus require to be executed in a specified order. In execution of such tasks the locality of data also becomes very important.
 - *Data Locality:* The data locality becomes of significant importance for scheduling, concurrent processing, and query processing in a large scale distributed/ Grid environment. In the temporal locality, the tasks requiring the same data set are scheduled to a single computational node. Whereas, in special locality, a task is scheduled on a computational node that is near to a data resource hosting the required data.
- **Resource Sharing Policies:** The resource sharing policies define sharing and access rules for software and hardware resources in the distributed environment. Such policies affect design of data organization and in turn the whole data management. In the Grid environment, such policies are usually implemented through Virtual Organizations [29]. The resource sharing policies can be broadly divided into the following three categories.

- *Regulated:* Under regulated resource sharing, there is a single entity (organization or individual) that forms the whole environment and defines the rules to access the shared resources. The other entities are bound to follow the rules set by the single entity.
- *Collaborative:* In collaborative resource sharing, the involved entities jointly set the data access rules and protocols.
- *Economic:* Under economic sharing the entities share resources for business purpose, whereby, each entity sets its own rules to access its shared resources. In such an environment the vision is usually to make profit.

IV. TAXONOMY OF DATA MANAGEMENT MODELS IN DISTRIBUTED/GRID ENVIRONMENT

There are eight DM models found in the existing distributed environment. These models are shown in the Figure 3. These models vary with respect to approaches they follow to do the DM tasks in the DM model (see Section 3.1). We describe these models in the following subsection.

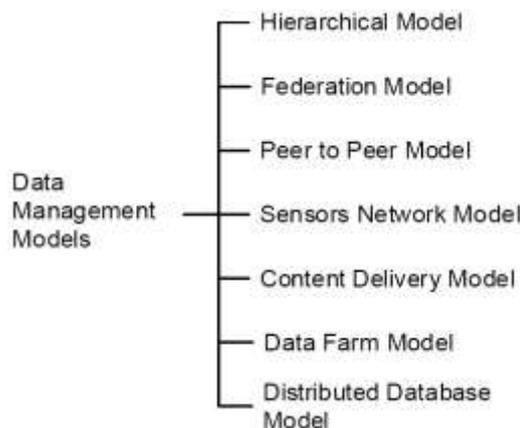


Fig.3. Taxonomy of data management models in distributed and Grid environments

4.1 Hierarchical Model

The Hierarchical model is commonly used in a distributed environment when there is a single data resource but the data needs to be distributed globally. A Hierarchical model is depicted in Figure 4, in which the requirements for distributing the data to various groups around the world are described. The tier-1 of this model, the Grid, deals with computation and storage of data. This data is then propagated to the Nodes distributed worldwide (continents/ regions/ countries), referred to as tier-2. From these nodes the data is passed down to the institutions (tier-3) and individuals (tier-4) who actually require it for processing. The tier-1 and tier-2 centers have to satisfy bandwidth, computational and storage requirements. Figure 5 depicts a self-explanatory Taxonomy of DM model components in the Hierarchical data management model.



Fig.4. The five layers of Hierarchical data management model

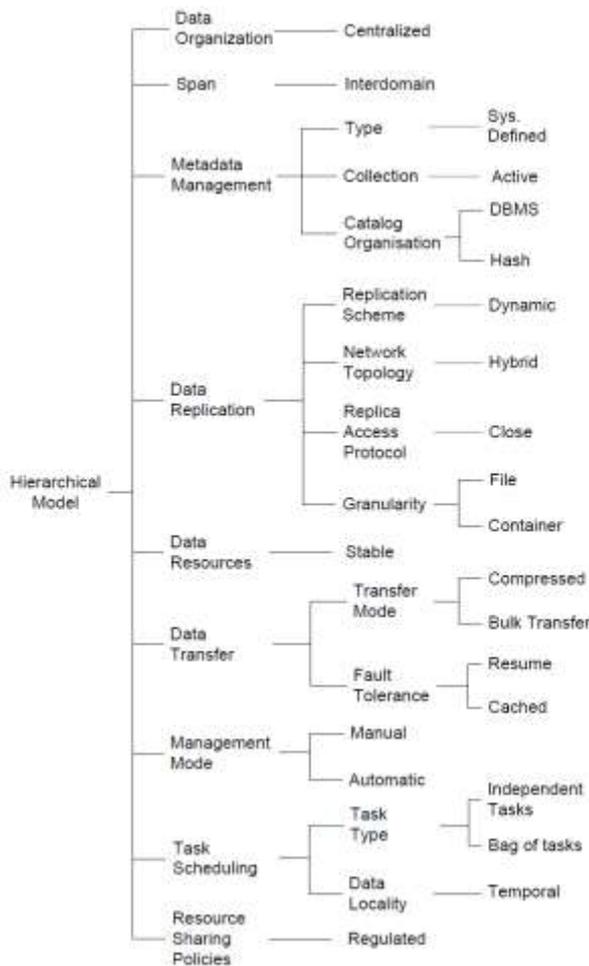


Fig.5. Taxonomy of DM model components in Hierarchical data management model

4.2 Federation Model

This model best suits in the situation where an existing collection of autonomous and possibly heterogeneous databases are required to be shared [41]. After authentication, the user can request data from any one of the databases within the federation. The owner institution has full control over local databases and has liberty of choosing among different configurations for different levels of autonomy, degree of replication, and cross-registration of each data resource. Figure 6 portrays the

Federation model and a self-explanatory Taxonomy of DM model components in Federation model is depicted in Figure 7.

4.3 Sensor Network Model

In this model the flow of data is from bottom to top where at the bottom lays sensors and at the top resides database. The data is collected from the sensors which are distributed worldwide. This data is gathered at a centralized place and is made available through a centralized interface, such as web portals etc. The

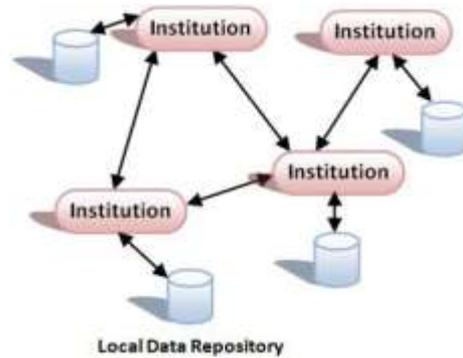


Fig.6. Federation data management model

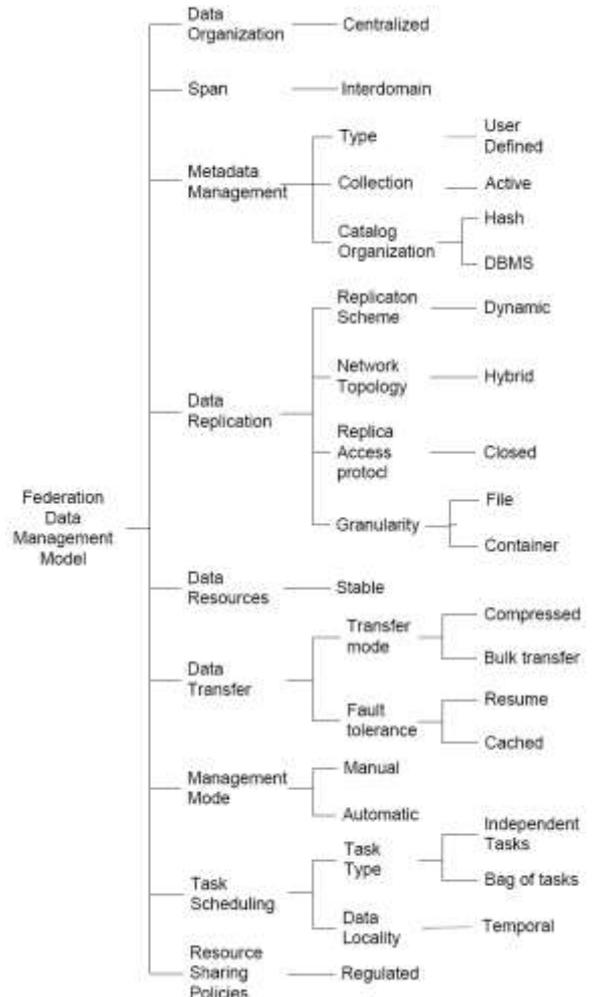


Fig.7. Taxonomy of DM model components in Federation data management model

centralized interface also facilitates verification and authentication. An architecture of Sensor Network model is shown in Figure 8. Figure 9 describes a self-explanatory taxonomy of DM model components in Sensor Network data management model.

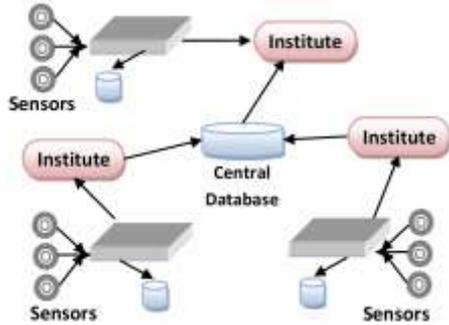


Fig.8. Sensor Network model with centralized storage

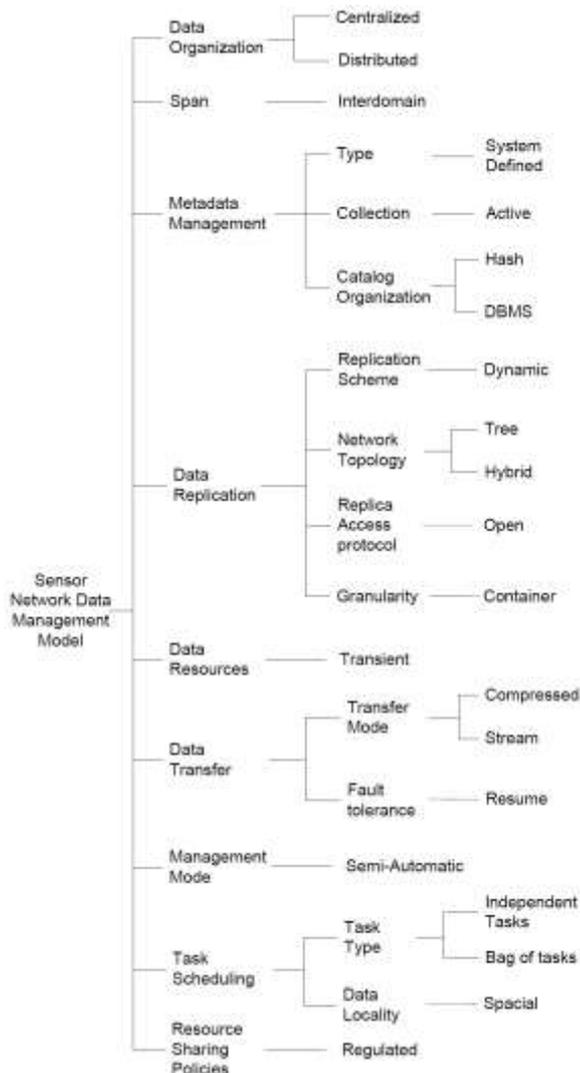


Fig.9. Taxonomy of DM model components in Sensor Network data management model

4.4 Peer-to-Peer Model

The Peer-to-Peer (P2P) model is based on ad-hoc aggregation of resources and the aim is to construct a

decentralized system [35, 22, 17]. In Peer-to-Peer model every node can be a server at one time and a client at some other time. The node behaves as server when it fulfills a request, and as a client when it sends a request thus resulting in a system that is decentralized so far as the control is concerned. Structure of P2P based data management model for Grid environment is described in Figure 10. Figure 11 shows a self-explanatory Taxonomy of DM model components in P2P data management model. The main focus of P2P model is on scaling, dynamicity, autonomy and decentralized control. Like the majority of other data management models, data replication, data storage, data management and security management are a part and parcel of this model but its

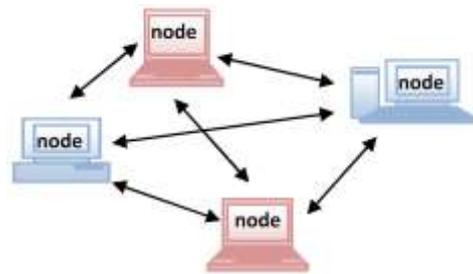


Fig.10. Peer-to-Peer model

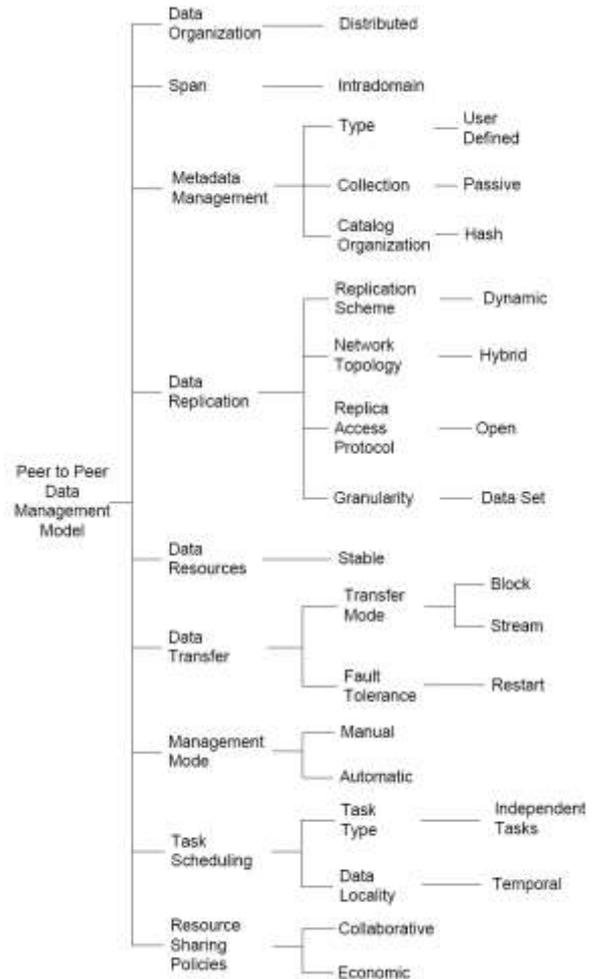


Fig.11. Taxonomy of DM model components in peer-to-peer data management model

data management capabilities are not sufficient for the Grid environment. P2P model is required to provide other functionalities, necessary in Grid environment, such as autonomic management, semantic based content and resource discovery, and workflow execution support.

In classical data Grid environments, usually data and data resources are centrally controlled by the owners. But, with increasing complexity of the systems, there is need of aggregation of data resources on ad-hoc basis. This need gives birth to structured P2P data management model for Grid environment [42]. In this model, data is organized as a multi-branch tree and the nodes are classified into three types: Global Grid Node(GGN), Local Grid Node(LGN)and Normal Grid Node(NGN). GGN and LGN manage the Grid, provide and consume storage resources, whereas NGN mainly consume storage resources [42].

4.5 Grid Datafarm Model

In Grid datafarm [44] model the data is replicated at each data resource of the Grid independently and in parallel by coupling storage, I/O bandwidth and processing. The data resources have a large disk space along with computational power. The nodes are connected with high speed connections such as Ethernet. The file system of Grid Datafarm is called Gfarm, which is a unifying file system in a sense that the name addressing space for files is throughout the Grid. The files in this systems are very large, usually in terabytes (TB), which are stored on multiple data resources in fragments, where individual fragments can be replicated.

Overall, the main focus of this model is large scale data which can be accessed with high speed in a very tightly coupled fashion. The architecture of this system is presented in Figure 12. Figure 13 shows a self-explanatory Taxonomy of DM model components in datafarm data management model.

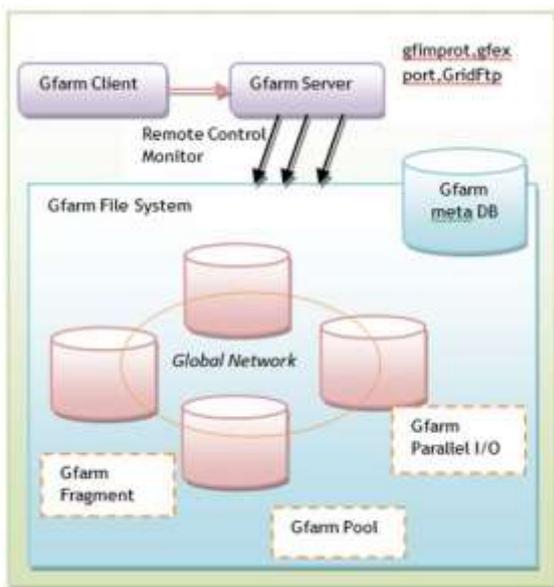


Fig.12. Grid Datafarm model (adopted from [44])

4.6 Content Delivery Model

In Content Delivery model there exist non-origin servers which work on the behalf of origin-servers [25, 26, 33]. The main server routes the client's request for processing to another server, called edge server; there are different edge servers in the network. So any client's request is directed to the one which is the closest to that particular client. The edge server caches the information/content to reduce the network traffic. If the edge sever contains the required data it fulfills the client's request directly. Otherwise, it gets the data from another edge server or the origin-server. The focus of this model is load balancing. It is very useful in case when data size is very large, such as images or video streams. The Taxonomy of DM model components in Content Delivery data management model is shown in Figure 14.

4.7 Distributed Database Model

It is a logical organization of distributed data resources, capable of executing local as well as global applications [38, 41, 30]. Such a model can be implemented either by splitting an already existing database, and distributing it on different nodes or federating it to provide a common

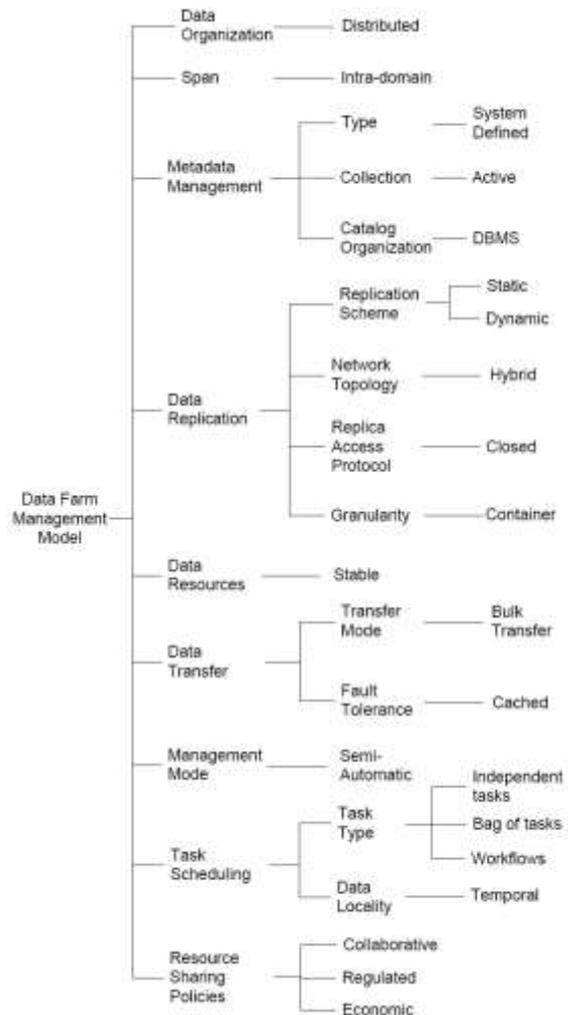


Fig.13. Taxonomy of DM model components in Grid Datafarm model

interface for access. The Distributed Database model requires various autonomy levels for tightly coupled databases nodes to completely independent ones. The model facilitates to:

- Remove centralized computer system
- Interconnect existing systems
- Replicate databases
- Distribute transaction processing
- Add new database for newly entered organization
- Distribute query organization
- Manage resource management efficiently

However there are a lot of strict requirements of this model, such as autonomy, consistency, isolation and durability, due to which it is not possible to implement this model in its true spirit.

The distributed environment/Grid needs to evolve more to implement this model in its current form. The Taxonomy of DM model components in Distributed Database management model is shown in Figure 15.

V. DATA MANAGEMENT MODELS IN THE EXISTING GRIDS ENVIRONMENTS

There are numerous existing and ongoing Grid projects from different domains. Usually, the Grid projects from scientific domains are data intensive. These projects are more often from domain of high energy physics, astronomy, bioinformatics and earth sciences etc. Generally, it can be said that the projects related to high energy physics use the data management model which is Hierarchical. The most famous high energy physics projects discussed extensively by Bunn and Newman [18] are WLCG [20], EGEE [6], EGI [7], GridPhyn [8], PPDG [14] and GridPP [10]. EGEE falls in the domains of biomedical science and uses Hierarchical data model. The major advantage of this model is that maintenance of consistency is pretty simpler due to its single source for the data. Biomedical Informatics Research Network (BIRN) [3] uses the Federation model for their data management in the Grid. This project belongs to the domain of bioinformatics and enables collaborations in

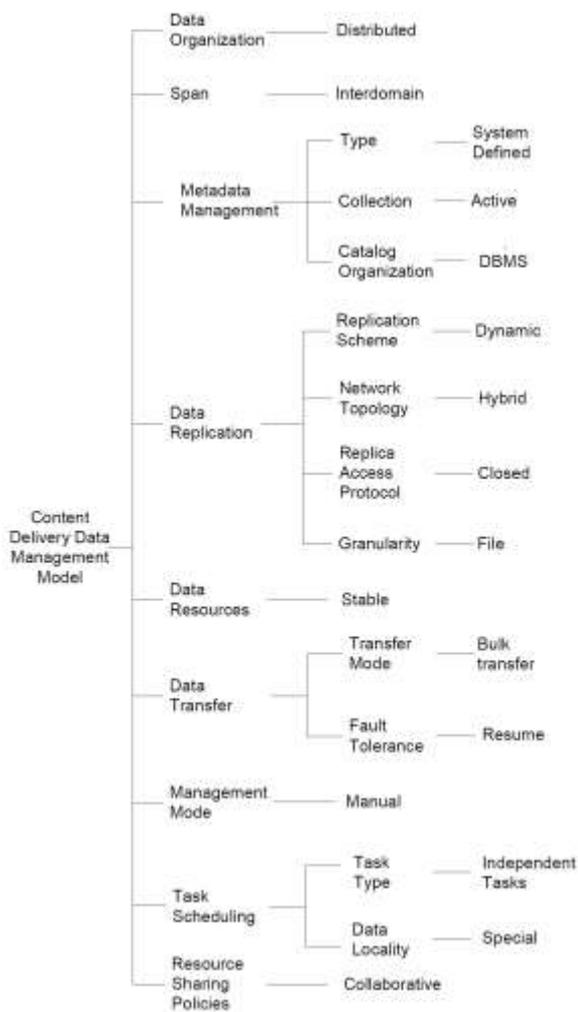


Fig.14. Taxonomy of DM model components in Content Delivery model

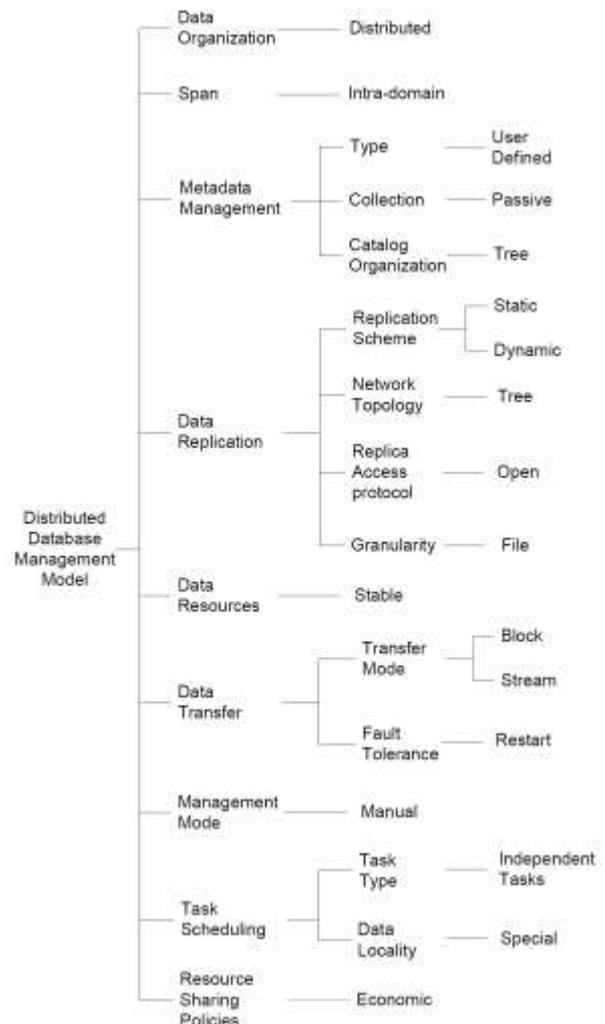


Fig.15. Taxonomy of DM model components in Distributed Database model

biomedical science through data sharing. BioGrid Japan [2] also uses Federation model for their data infrastructure. Virtual observatories [11, 1] has also adopted Federation model for managing the access to astronomy observations and simulation archives. Earth System Grid [4] has integrated computational data and resources to create an infrastructure for climate research through Federation model data management. eDiaMoND [5] works on cancer treatment and provides access to their Distributed Database of mammo-gram images. They have also integrated their system through Federation model.

In Sensor Network model, data is collected from the sensors, which are distributed worldwide to collect information. In this model, the nodes at the bottom of the hierarchy are the original sources of the data which is stored at a central repository. This model is adopted by NEESgrid [12, 39]. It helps the scientist to perform their experiment at distributed nodes and then evaluate data through a consistent interface. This model differs from others due to single access point of data, whereas in other models data can be accessed from multiple data resources. This model best suites the scenario where the overhead of data replication decrease efficiency and access of data for some particular regions.

The Real-time Observatories, Applications, and Data Management Network (ROADNet) [36] uses Hybrid data model approach of Sensors and Federation data models. The data is collected from sensors and is registered in the Grid using object ring buffer. These buffers are further federated to form virtual object ring buffer. Narada [23], P-Grid [15], Grid4All [9] and Organic Grid [13] are using P2P data management model. This trend of peer-to-peer is getting more and more popular; the Global Grid Forum has recently created the OGSA-P2P group [19] to extend OGSA for the development of P2P applications. New advancements in Grid computing and its relationship with other technologies (Web Services, P2P, Agents) has broaden the horizon of data management models in the Grid. A list of some of the existing projects, their domain, and data models they employ is provided in Table 1.

VI. CURRENT CHALLENGES AND FUTURE WORK

In an environment where applications need heavy computations on geographically distributed resources and also require collaboration and sharing, data and its management becomes a very significant issue, as ultimately it is all about data. There is a lot to do in the distributed environment regarding data management. Currently there is no such data management model which is autonomous. To facilitate it, there is a need to add semantics in data management. As there exist the most sophisticated types of data management systems in traditional environment, such as relational database management system (RDBMS), object relational database management systems (ORDBMS) etc. it would be very beneficial to incorporate such types of models in distributed environments like Grid [46]. A robust data management model is also missing in the Grid

Table 1. Data management models in some of the existing Grid environments

Project	Domain	Data Model
LCG	High Energy Physics	Hierarchical
EGEE	High Energy Physics	Hierarchical
PPDG	Physics	Hierarchical
GridPhyn	High Energy Physics	Hierarchical
Grid3	Physics, Biology	Hierarchical
BIRN	Bio-Informatics	Federated
GridPP	High Energy Physics	Hierarchical
BioGrid	Protein Simulation,	Federated
ESG	Climate Modeling	Federated
BADG	High Energy Physics	Hierarchical
NESSGrid	Earthquake Engineering	Monadic/Sensors
ROADNet	Real observation	Time Hybrid
UK Science	e-Scientific Research	Federated
IVOA	Astronomy	Federated
EDiaMond	Breast Cancer Treatment	Federated
Grid4All	Democratic Grid	P2P

environment. With the help of AI (Artificial Intelligence) the Grid's data management can become more powerful and robust. A complementary practice of agents in Grid is also an open issue which needs to be explored [28].

Besides the factors described above there are a lot of functions which are open issues and should be focused for future research. As in the Grid, there exist different types of data; from heavy video streams to very light-weight devices' (mobile etc.) data, there is need to have different management techniques to handle different types of data. Data security is still an issue which needs to get addressed deeply. Although different models provide data security but still there is need to have a fool-proof data security system.

In the traditional data management systems, the query processing deals with single data resources or a limited distributed database systems, but the dynamic environments like Grid, where there are multiple heterogeneous resources, require a special support for distributed query processing.

VII. CONCLUSION

In this paper, a brief overview of data, data management systems, data management models and their taxonomy, and existing Grid environments in the context of their data management models is presented. The different models of data management are classified due to the differences in their architecture or topologies. It is required from these models to address the issues related to data management in such a fashion that the needs of distributed/Grid environments are satisfied. There are several tasks of data management in the Grid among which schema management, data resource discovery, query processing, load balancing, replication, caching, workflow management, autonomic data management and security can be defined as the core ones. The different

models satisfy these core functionalities partially or almost completely. The table 2 below describes the extent

to which the presented models can fulfill these managerial requirements.

Table 2. A comparison data management models for their support for major data management tasks

	Hierarchical Model	Federation Model	Sensor Model	Detailism Model	Distributed Database Model	Content Delivery Model	Peer-to-Peer Model
Data Transparency & Consistency	++	++	++	++	++	++	++
Autonomy Support	-	-	++	+	+	++	+
Resource Discovery	++	++	++	++	++	++	++
Query Lucidity	++	++	++	+	+	+	++
Efficiency	++	++	++	++	++	++	++
QoS	++	+	+	++	+	++	+
Fault Tolerance	++	++	+	++	++	+	++
Work Flow Support	++	++	++	++	++	++	++
Security	++	++	++	++	++	++	++
Data Resources	++	++	++	++	++	++	++
Metadata	++	++	++	++	++	++	++
System Information	++	++	++	+++	+++	+++	+++
Extensibility	++	++	++	+++	+++	++	+++
Does not exist		-					
Up to some extent		+					
Fairly done		++					
Good		+++					

There are still a lot of aspects to explore and analyze. In this regard future work is also suggested which can lead to discover more horizons

REFERENCES

- [1] Australian Virtual Observatory. <http://www.aus-vo.org/>.
- [2] BioGrid, Japan (biogrid-jp). <http://www.biogrid.jp/>.
- [3] Biomedical Informatics Research Network (BIRN). birncommunity.org/.
- [4] Earth System Grid. www.earthsystemgrid.org/.
- [5] eDiaMoND Grid Computing Project. www.ediamond.ox.ac.uk/.
- [6] Enabling Grids for E-Science (EGEE). <http://public.eu-egee.org/>.
- [7] European Grid Infrastructure (EGI). <http://www.egi.eu/>.
- [8] Grid Physics Network (GridPhyN). <http://www.gridphyn.org/>.
- [9] Grid4All. <http://www.cslab.ece.ntua.gr/cgi-bin/twiki/view/CSLab/AboutGrid4All>.
- [10] GridPP: UK Computing for Particle Physics. <http://www.gridpp.ac.uk/>.
- [11] International Virtual Observatory Alliance. <http://www.ivoa.net/>.
- [12] NEESgrid: Virtual Collaboratory for Earthquake Engineering. www.neesgrid.org.
- [13] Organic Grid. <http://www.csc.lsu.edu/~gb/OrganicGrid/>.
- [14] Particle Physics Data Grid (PPDG). <http://www.ppdg.net/>.
- [15] Karl Aberer, Philippe Cudr  -Mauroux, Anwitaman Datta, Zoran Despotovic, Manfred Hauswirth, Magdalena Puceva, and Roman Schmidt. P-grid: a self-organizing structured p2p system. SIGMOD Record, 32(3):29–33, 2003.
- [16] Carlisle Adams and Steve Lloyd. Understanding the Public-Key Infrastructure: Concepts, Standards and Deployment Considerations. Addison-Wesley Professional, second edition.
- [17] David P. Anderson, Jeff Cobb, Eric Korpela, Matt Lebofsky, and Dan Werthimer. Seti@home: an experiment in public- resource computing. Commun. ACM, 45(11):56–61, November 2002.
- [18] Fran Berman, Geoffrey Fox, and Anthony J. G. Hey. Grid Computing: Making the Global Infrastructure a Reality. John Wiley & Sons, Inc., New York, NY, USA, 2003.
- [19] Bhatia, K.I. Ogsa-p2p research group: peer-to-peer requirements on the open grid services architecture framework. Global Grid Forum Document GFD-I.049, 2005.
- [20] CERN: European Council for Nuclear Research. Worldwide LHC Computing Grid (WLCG). <http://wlcg.web.cern.ch/>.
- [21] Ann Chervenak, Ian Foster, Carl Kesselman, Charles Salisbury, and Steven Tuecke. The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets. JOURNAL OF NETWORK AND COMPUTER APPLICATIONS, 23:187–200, 1999.
- [22] Ding Choon-Hoong, Sarana Nutanong, and Rajkumar Buyya. Peer-to-Peer Networks for Content Sharing, chapter 2, pages 28–65. Idea Group Inc, Hershey, PA, USA, 2005.
- [23] Community Grids Lab, Indiana University. The Narada Brokering Project. <http://grids.ucs.indiana.edu/ptliupages/projects/narada/>.
- [24] Data Management International. DAMA-DMBOK Guide (Data Management Body of Knowledge) Introduction & Project Status. http://www.dama.org/files/public/DI_DAMA_DMBOK_Guide_Presentation_2007.pdf.
- [25] Brian D. Davison. A web caching primer. IEEE Internet Computing, 5(4):38–45, July 2001.
- [26] John Dille, Bruce Maggs, Jay Parikh, Harald Prokop, Ramesh Sitaraman, and Bill Weihl. Globally distributed content delivery. IEEE Internet Computing, 6(5):50–58, September 2002.
- [27] David F. Ferraiolo, D. Richard Kuhn, and Ramaswamy Chandramouli. Role-Based Access Control. Artech House.
- [28] Ian Foster, Nicholas R. Jennings, and Carl Kesselman. Brain meets brawn: Why grid and agents need each other. In Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1, AAMAS '04, pages 8–15, Washington, DC, USA, 2004. IEEE Computer Society.

- [29] Ian Foster, Carl Kesselman, and Steven Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *Int. J. High Perform. Comput. Appl.*, 15(3):200–222, August 2001.
- [30] Jim Gray and Andreas Reuter. *Transaction Processing: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition, 1992.
- [31] Wolfgang Hoschek, Francisco Javier Ja én-Mart ínez, Asad Samar, Heinz Stockinger, and Kurt Stockinger. Data management in an international data grid project. In *Proceedings of the First IEEE/ACM International Workshop on Grid Computing, GRID '00*, pages 77–90, London, UK, UK, 2000. Springer-Verlag.
- [32] Bart Jacob, Luis Ferreira, Norbert Bieberstein, Candice Gilzean, Jean-Yves Girard, Roman Strachowski, and Seong (Steve) Yu. *Enabling applications for grid computing with globus*. IBM Corp., Riverton, NJ, USA, first edition, 2003.
- [33] Balachander Krishnamurthy, Craig Wills, and Yin Zhang. On the use and performance of content distribution networks. In *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement, IMW '01*, pages 169–182, New York, NY, USA, 2001. ACM.
- [34] Jagatheesan A. Rajasekar A. Wan M. "Moore, R.W. and W" Schroeder. Data grid management systems. In the 21st IEEE/NASA Conference on Mass Storage Systems and Technologies (MSST), College Park, Maryland, USA, April 2004.
- [35] Andy Oram, editor. *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*. O'Reilly & Associates, Inc., Sebastopol, CA, USA, 2001.
- [36] John Orcutt and Frank Vernon. *Real-time Observatories, Applications, and Data Management Network [ROADNet]*. <http://roadnet.ucsd.edu/>.
- [37] Esen Ozkarahan. *Database management: concepts, design, and practice*. Prentice Hall, 1990.
- [38] M. Tamer Ozsu. *Principles of Distributed Database Systems*. Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd edition, 2007.
- [39] Laura Pearlman, Carl Kesselman, Sridhar Gullapalli, B. F. Spencer, Jr., Joe Futrelle, Kathleen Ricker, Ian Foster, Paul Hubbard, and Charles Severance. Distributed hybrid earthquake engineering experiments: Experiences with a ground shaking grid application. In *Proceedings of the 13th IEEE International Symposium on High Performance Distributed Computing, HPDC '04*, pages 14–23, Washington, DC, USA, 2004. IEEE Computer Society.
- [40] Jun Qin and Thomas Fahringer. Advanced data flow support for scientific grid workflow applications. In *Proceedings of the 2007 ACM/IEEE conference on Supercomputing, SC '07*, pages 42:1–42:12, New York, NY, USA, 2007. ACM.
- [41] Amit P. Sheth and James A. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Comput. Surv.*, 22(3):183–236, September 1990.
- [42] Wei Song, Yuelong Zhao, Wenying Zeng, and Wenfeng Wang. Data grid model based on structured p2p overlay network. In *Proceedings of the 7th international conference on Advanced parallel processing technologies, APPT'07*, pages 282–291, Berlin, Heidelberg, 2007. Springer-Verlag.
- [43] William Stallings. *Cryptography And Network Security: Principles and Practices*. Prentice Hall.
- [44] Osamu Tatebe, Youhei Morita, Satoshi Matsuoka, Noriyuki Soda, and Satoshi Sekiguchi. Grid datafarm architecture for petascale data intensive computing. In *Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid, CCGRID '02*, pages 102–, Washington, DC, USA, 2002. IEEE Computer Society.
- [45] The Globus Project. Globus toolkit. <http://www.globus.org/toolkit/docs/latest-table/gridftp/>.
- [46] Srikumar Venugopal, Rajkumar Buyya, and Kotagiri Ramamohanarao. A taxonomy of data grids for distributed data sharing, management, and processing. *ACM Comput. Surv.*, 38(1), 2006.
- [47] Srikumar Venugopal, Rajkumar Buyya, and Kotagiri Ramamohanarao. A taxonomy of data grids for distributed data sharing, management, and processing. *ACM Comput. Surv.*, 38, June 2006.

Author's Profile



Farrukh Nadeem received his Ph.D. in 2009 from the University of Innsbruck, Austria. Since July 2009, he has been working as an Assistant Professor. His main research interests include performance modeling and prediction, and scheduling scientific workflows in distributed systems, particularly the Grid and the Cloud. He has been involved in a several Austrian and Saudi research and development projects. Farrukh is author of more than 24 papers, including four book chapters. He holds several distinctions and awards during his educational career.

How to cite this paper: Farrukh Nadeem, "A Taxonomy of Data Management Models in Distributed and Grid Environments", *International Journal of Information Technology and Computer Science(IJITCS)*, Vol.8, No.3, pp.19-32, 2016. DOI: 10.5815/ijitcs.2016.03.03