ICES/KIS Project No. 1544516

Title: **The Grid information System**

Authors: **Adam Belloum**

Date: **18th April 2001**

Document Code: **/UvA/VLAM-G-AM/TN04**

Version: **1.1**

Status: **Proposal**

Technology & Scientific Center
WATERGRAAFSMEER

The Grid information System
(Survey)

Adam Belloum

document version 18th April 2001

Note to the readers:

- The status of this document is a proposal, *i.e.* it may be considerably changed in the future due to remarks of its readers. It provides a starting point for discussion on the design of the VLAM-G-AM RTS and may point to future challenges.
- This report aims also at providing enough references needed for the implementation of the VLAM-G-AM RTS.
- While reading this short report the reader is invited to give his comments and remarks. We especially welcome comments from VLAM-G developers on how they see their developments integrated in the VLAM-G AM-RTS, or which services they need from the VLAM-G AM-RTS.
- Notation: if *?? sentence/word ??* is found in the text it means that this sentence/word might be not appropriate to describe the idea being discussed.
  The color used in the figures represent different components
  - Yellow: Globus toolkit components;
  - Red: packages developed for Globus;
  - Gold: Database components;
  - Light Blue: Module developed
  - Brown: Module to be developed
  - Cyan: shared elements, domains etc.;
  - Green: Potential software for the implementation;
- Web pages where the reader can find more in formation on the VLAM-G Abstract Machine developments:
  - VLAM-G-AM Module skeleton API User Guide:
    http://www.science.uva.nl/~adam/psfiles/VlAPI.ps
  - http://www.nikhef.nl/~davidg/vlab/
  - http://www.science.uva.nl/~zegerh/work.html
  - http://www.science.uva.nl/~adam/currentResearch.html
- This document is the result of VLAM-G-AM design meetings. During this project, many people have been contributed. This makes it almost impossible to mention all of them. However, we do want to name D. Groep, C.R. Garite and E. Kaletas.

Abstract

The grid technology is an emerging technology, it aims at providing services to ease the construction of distributed systems involving heterogeneous components, located at different organizations and using various security infrastructure. At the heart of this technology stands a challenging problem related to the modeling of information contained in the sets of data considered in grid-applications. So far, an extensive work has been done to provide grid-users with services allowing them to represent and manipulate files containing raw data. However, this does not seem to fulfill the requirements of a number of grid-application. The concept of the data-grid introduced to deal with the data-manipulation on the grid has to be extended. It is the goal of this report to study the methods and techniques needed to support the representation of the information contained in the data sets. The ultimate goal of this study is to define a model for representing a information on the grid and the necessary tools to integrate this model within the grid-technology.

1 Introduction: Requirements & Challenges

In Grid Technology, data manipulation is one of the most challenging problems. All grid applications developed so far have shown intensive use of large data sets. It is therefore important to provide scalable solutions for the manipulation of the data in grid computing. Because of the distributed character of the grid, data has to be accessed remotely by specific communities, stored on geographically distributed and/or federated locations, transfered among heterogeneous systems, converted to different formats, etc.

Some the data manipulation within the grid is far from trivial; the research groups working on the access to data in grid applications are still investigating a number of approaches to define the appropriate architecture for data manipulation of what is now known as the "Data-Grid".

Reagan Moore defines the Data-Grid as a distributed system that ties together data and computing resources [12]. This definition has added the computing resources as part of the data manipulated within the grid. A non-exhaustive list of requirements has already been proposed and discussed within the grid community [22, 17]. The Data-Grid is a middle-ware layer on which the developers of grid applications can rely to support all aspects of the data access and data manipulation. So far, the Data-Grid systems have been described as systems providing the following services:

- Data handling: read data from a remote resource
- Remote processing: filter before transmission over the network
- Publication: add data sets to collections for use by others
- Information discovery: query across multiple repositories
- Analysis: ability to use data in simulation, data mining, figures.

One of the most important Data-Grid requirements is the ability to support location-independent data access, *i.e.*, the grid user does not have to deal with the low-level grid details. Five level of "transparency" have been identified by the Data Access Working Group (a working group within the Grid forum), each transparency layer hides a specific grid implementation details:

- Name transparency (important for the information discovery): users should be able to identify appropriate data sets by lists of attributes. Name transparency provides a useful abstraction for heterogeneous data access. Still an information model is needed to organize the attributes. Two models are currently being used:

  - Keyword/attribute as used in LDAP
  - DTD used in XML An ongoing discussion within the grid forum is trying to define a description of objects within the grid (a Grid Object Specification or GOS). A current implementation uses the Meta-computing Directory Services Markup Language (MDSML). The GOS will have binding to XML [21].

- Location transparency: hides the physical location from the users or the type of software storage system used to control the data storage peripheral. The actual location of the data set will be maintained as one attribute of the data set.

- Protocol transparency: the data handling system will use attributes stored in the information discovery catalog to determine the particular access protocol required.

- Time transparency: deals with the reduction of data access time to achieve high performance. To achieve this goal techniques like replication, prefetching, and parallel I/O are considered.

- Single sign-on transparency: this layer of transparency allows single sign-on for all security domains comprising the grid. Users will be required to sign-on only once per session.

Current Data-Grid systems rely on “Information Discovery Catalogs”. These catalogs can be queried to obtain information both on the available (software or hardware) resource and on the data. Information Discovery Catalogs are implemented by means of directory services like LDAP, relational and object-oriented databases, and even Unix files.

The data management introduced by the Data-Grid is focusing mainly on file manipulations as was pointed out by Roy Williams in his comments on the paper describing the architecture of the Data-grid [5]. The main argument of Roy Williams focuses on the fact that services that going to use the grid require more advanced data access.

2 The Architecture Data-Grid as proposed within the Grid forum

The current architecture for the Data-Grid data access is composed of a collection of services that provide protocol, location, and time transparency. A glue layer provides the access interface to each of the capabilities of the data access system: the “Data Handling System”(Figure 1).

There are already some implementations of (at least parts of) this Data-Grid architecture, such as Condor [18], NILE [10], SRB [3], and GASS [4]. In this architecture all information needed to access the data is provided by the module called *Information discovery*, which, when implemented, is often a hierarchical tree based on the LDAP directory service.

According to this architecture the *Information discovery* module get its input mainly from the low level module which provides information on grid resources. For data sets manipulated by the grid applications, the data handling system identifies data sets using metadata catalogs. The type of attributes necessary for data-set discovery and their format are still being discussed within the Information Management GridForm working group. Among the types and formats that have been already discussed:

Dublin core : for provenance or origin information

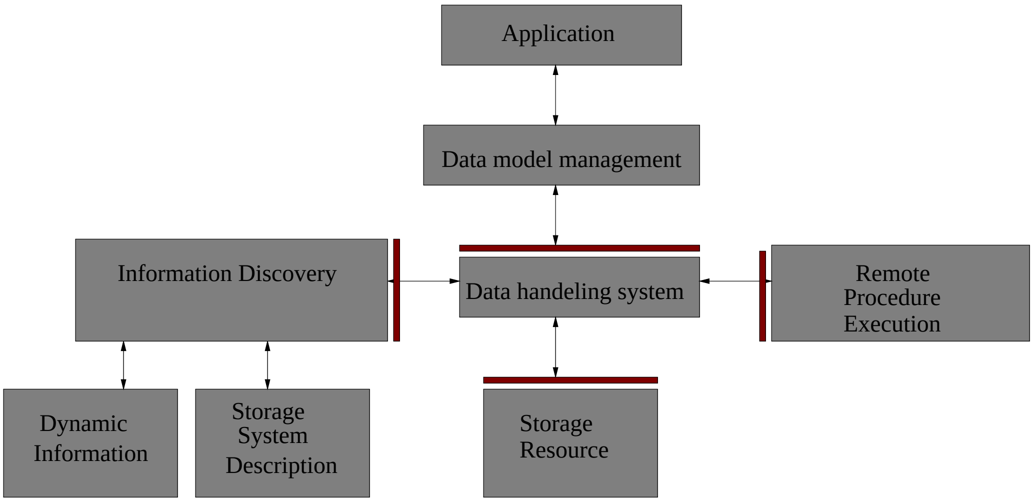

Figure 1: *data handling systems*

'Unix level' file attributes : for tracking changes to a data set, access control, audit trails figures.

Structural attributes : data format, geometry, physical units of variables, feature annotation.

Domain specific attributes : context of a data set with respect to the collection domain.

Information related to the attributes such as the organization of the attributes, the associated semantics, etc., needs to be published. The information discovery needs a standard form to describe the organization of the data and the XML DTD format is such a potential standard in the near future.

The aim of this architecture is to tackle interoperability problems between the legacy infrastructure components. The four interfaces outlined in Figure 1 deal with these interoperability problems.

3 Data-Grid Implementations

A number of data handling systems have been already developed and can be used in grid applications. A framework for the comparison of these systems has been proposed by the Data Access Working Group [17]. It focuses on six issues:

1. Accessing entire file:
    - Support for multiple disk storage systems such as UNIX, DPSS, databases
    - Access to data storage on tape such as HPSS, Unitree, ADMS
    - Transfer of individual files, including generic services such as FTP, HTTP, and SCP, and custom services such as GASS and GSS-FTP
    - Transfer of aggregates of files
2. Accessing parts of a file:
    - Invocation of common filtering functions to extract a subset of a file.
    - Invocation of remote procedures to provide a generalized file filtering capability.
    - Direct application access to block or byte strings of data

3. Support for distributed data storage: includes support for replicas across a WAN, naming mechanisms, management of aggregates of files

4. Information discovery services: management of metadata at both application and system level, discovery of metadata to optimize performance.

5. Authentication services: authentication of users across administrative boundaries, authentication of resource managers acting on behalf of users

6. Reservation services: cache and disk space allocation, access to drives and media in tape storage systems, network bandwidth and quality of service.

Using this framework the Data Access Working Group has compared Globus, the Storage Request Broker(SRB), Legion, and Internet Backplane [17]. It appeared from this comparison that Globus is the only toolkit that provides, to some extent, all six services listed in the framework. The Globus toolkit has the advantage of supporting the resource reservation services through the GARA (Globus Architecture for Reservation and Allocation). This comparison shows only if the toolkit does or does not support the services, it does not include any performance test of the services provided by the toolkits.

Another comparative study uses the transparency layers to compare a number of data handling systems [22]. Table 2 shows how each toolkit deals with each level of transparency.

Data handling sys.	Name Trans.	Location Trans.	Protocol Trans.	Time Trans.
SRB - Storage Resource - Broker	metadata catalog of attributes	Location specified as attributes	Access protocol specified as attributes	Replication containers for data aggregation
GASS - Globus Archival - Storage Service	File name	URL	Access to remote file system	Prefetch of data sets
DPSS - Distributed Parallel - Storage System	File name	Common directory	Access to multiple DPSS servers	Parallel I/O from DPSS cache
Internet-2	File name	Common directory		Replication in multiple caches
Condor	File name ??	IP address ??	Access to remote file systems	
Legion	Object ID	Class specification of vault	Access to Legion vaults	Migration between vaults
ADR - Active Data - Repository	File Name ??*			composition and data placement
NILE	Objects	Virtual File system		
DFS - Distributed File - System	File Name	Common directory	Global File system	Caching at local site

Table 1: *Workloads characteristics*

4 Handling Data Access within the VLAM-G. project

The application we foresee for the VLAM-G are typical examples of grid applications: huge sets of data have to be accessed and moved within a heterogeneous

*??: the author of the comparison is not sure about the statement

environment. The case studies we have been investigating are showing two types of data: on the one hand we have structured data stored in databases and accessed through queries, on the other hand we have non-structured data stored in files.

As all grid-applications, the VLAM-G has to deal with the problems related to the data access and manipulation. The integration of VLAM-G data within the grid infrastructure is straight forward for the non-structured data since they are just file and Globus (or any grid technology) provides a number of middleware primitives that can handle this kind of access to files with in relatively secure and efficient way.

However, for structured data the integration might pause some problems:

1. Information management, discovery services, and data 'publishing' based on structured data access, has not been extensively discussed in the Grid community. Although it has been stated that database access (primarily Oracle and Objectivity) should be 'supported', none of the available Grid tools, presented at the IPG and IS workshops, address these issues.

2. Even if tools are developed later on within the Globus context, these might have strict bindings to specific database systems. In any case, Matisse will likely not be supported explicitly, although 'generic' database access (ODBC?) might be an option.

For these two reasons, we have to develop the database interface to the VLAM-G architecture in particular and the grid in general ourselves. Integrating the VLAM-G data within the Data-Grid architecture has an advantage; this integration give access to all the tooling being developed for the grid computing: remote data access, data replication data caching, data reallocation, data discovery etc. This means that when developing the VLAM-G interface all this data aspect manipulation do not have to be re-considered.

	VLAM-G Approach	Globus Approach
Data Access	Application data is stored in a metadata managed by Matisse. System data (resources)	Both application and system data are accessed through metadata. The metadata is managed through the Meta-Computing Directory Service (MDS), implemented using LDAP Technology
Data Manipulation	to be discussed	A number of services have been developed and allow a number of data manipulation such as: WFS, NetSolve, GARA, figures

Table 2: *Workloads characteristics*

Since the VLAM-G is building a working environment for scientific experiments, it has to provide access not only to raw data but also to services that provide information extracted from raw data, existing databases, or any other information systems. It is likely that the architecture for the data handling, as proposed now (Figure 1), is not sufficient and another interface has to be considered. This interface has to hide the heterogeneity of the information systems, and to provide full control of the data set by its owner (*i.e.*, support different views when accessing the data). The VLAM-G data handling system is the natural extension of the architecture proposed by the Grid Forum. Figure 2 shows how the new module can be easily integrated within the architecture proposed in the grid forum. The new module allows mainly the integration for information coming from different sources. Such an interface requires advanced information management techniques that allow to support heterogeneous systems. Similar problems have been addressed extensively in the area of federated databases, virtual organizations, and digital libraries [7, 9, 2].

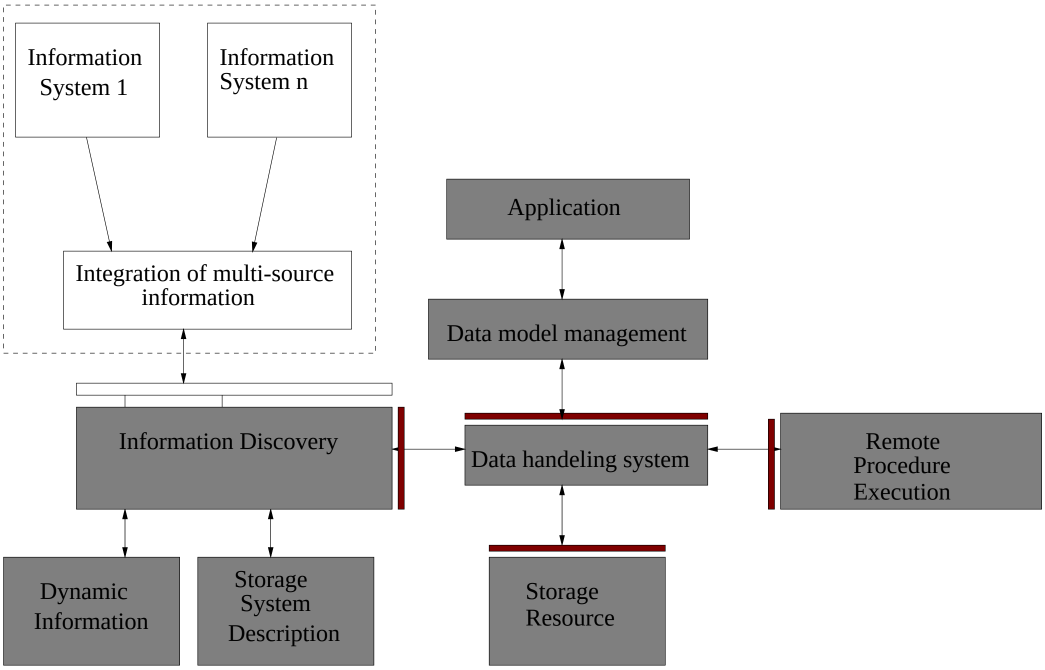

Figure 2: *Data handling system for VLAM-G*

Within the Grid forum, the information services working group is currently investigating the representation of data. The aim is to define a standard schema for the grid informations services. An object oriented approach is being discussed and a specification format called the Grid Object Specification (GOS) has been proposed. This format builds upon the syntax developed as part of the Globus/MDS project and uses the LDAP terminology [20]. A prototype of a tool for converting the GOS to some data standards have been already developed, the supported conversion currently being html, mdsml, dsml, and ldapcofig.

5 Data Content Manipulation

Data manipulation is one of the main concern of the grid-technology, they way the data is manipulated depends on how the data is regarded; if data is considered just as a set of bytes stored in files then the data-manipulation is restricted to providing and controlling the access to the data files. However, if the interest is focused on the information contained in the raw data then the data-manipulation has a wider sense. The most commonly used way to represent the information contained in the raw data, is the use of what is known as *metadata*. Tim-Berners described metadata as: "Metadata was called *metadata* because it started life, and is currently still chiefly, information about web resources, so data about data. In the future, when the metadata languages and engines are more developed, it should also form a strong basis for a web of machine understandable information about anything: about the people, things, concepts and ideas".

In practice metadata is composed of a set of attributes having each a particular type and capturing each a particular view of the information, the attribute value pair model is defining the semantics of the metadata. Five categories of attributes have been proposed to define a set of a standard attributes in [13]: identification, definition, relation, representation, and administrative". In digital information repository, metadata plays two important roles: on one hand it provides the exact location of the requested data, on the other hand it supplies information about the content of the data [15].

Metadata seems to become a consensus to represent the information contained in the data, remains to be sure that a model such as the attribute-values pair is the appropriate model to represent the information contained in the data; especially if we consider new types of data such as images, sound or multimedia. In the domain of image retrieval, even if the current images retrieval systems are text-based, the current research seems to promote content-based image retrieval where images are described in term of color, texture, and shape [1].

The concept of metadata modeled as a pair of attribute-value has been already used in the grid technology in the development of the grid information service (GIS). A Meta-computing directory service (MDS) implemented as an LDAP directory contains all the metadata describing the grid software and hardware resources. In the data-handling shown in Figure 2, the MDS is part of the information discovery module. A data model specification, called GOS based on the attribute-value pair allows to represent a compute resource on the grid (Figure 3), a person using the grid [6], etc.

```
GRID::ComputeResources IBJECT-CLASS ::= {
        DUBCLASS OF Grid::PhysicalResource
        RDN = hn (hostName)
        CHILD OF {
            Grid::organizationalUnit,
            Grid::organization,
        }
        MUST CONTAIN {
            canonicalSystemNames :: cis,
            manufacturer :: cis, single,
            model :: cis, single,
            serialNumber :: ces, single,
        }
        MAY CONTAIN {
            diskDrive :: dn, multiple,
        }
}
```

Figure 3: *GOS model for representing people on the grid*

The GOS has been proposed to represent mainly information related to the grid management. The data manipulated in the grid-application is still considered as a collection of files.

During 5th grid forum, Plale and Dinda presented a talk entitled "LDAP Query Access: Limitation and Opportunities" where they have pointed out among other points that LDAP query access maybe inadequate for the typical types of queries sent to the GIS [19]. The limitation of LDAP is mainly due to its hierarchical data model, the result of an inefficient query access could be quite dramatic to the performance of the whole grid more work performed at the server, useless data transfer, and increase the operations performed at the clients site. Plale and Dinda proposed two three approaches to solve the problem: influencing the current work on the extension of LDAP query language, representing the data using the relational data model, and finally adding a front-end interface to the GIS to do relational-style query access. The dynamic query Object (dQUOB) has been proposed as a possible way towards the solution. The dQUOB has been developed for querying for specific information from high-volume data streams [14]. The dQUOB consists of a compiler and a run-time system environment. The compiler transform the query into an intermediate form, and generates a script that it moved into a doublet called *quoblet* which consists of an interpreter to decode the script, and the dQUOB library. The script contains enough information to retrieve and dynamically link-in the action code. The run-time environment contains a re-optimizer that gathers statistics, and periodically triggers re-optimization.

With the explosion of the Internet new methods emerged in the field of the metadata manipulation. Programs known as spiders, knowbots, or automatic robots can extract metadata from objects there were made available on the Internet [1]. Even if these programs are far from effective they represent an interesting concept, that could fill the gap between the grid-applications on one hand and the grid-technology on the other hand. In the context of the virtual-laboratory generating on the fly the metadata of the results of complex and time consuming database requests can allow the creation of caching mechanism where information is stored (Figure 6).

6 Implementation Issues

Allowing data content manipulation on the grid is allowing to hookup easily databases on the grid. Database management system might be considered as a new type of resources on the grid with which grid-users and grid-tools can interact in a strait forward way.

6.1 Potential technologies

During the last decade a number of solutions have been proposed to allow querying database in a distributed and heterogeneous system. Specifications for database access such as ODBC, and JDBC are widely used, when combined with the CORBA object implementation, database access can become much easier in distributed and heterogeneous environment. Such a combination has been successfully used in the work of Pethuru and Naohiro to implement the interoperability of biological Databases [16]. The integration of this concept in the grid data handling system can be strait forward since the needed information about the CORBA Object Request Broker (ORB) can be stored in the resource discovery system (Figure 4).

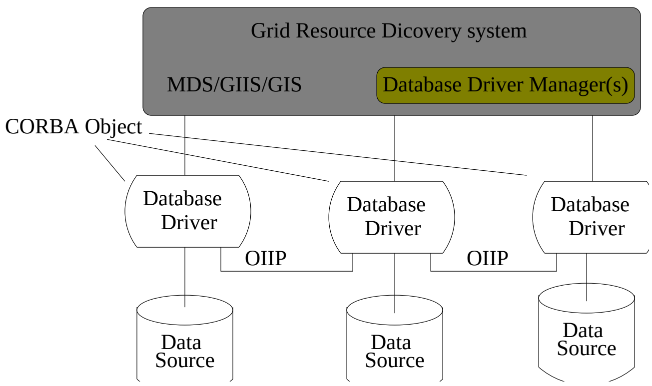

Figure 4: *using CORBA object to promote interoperability*

CORBA solves partially the problem of interoperability since a fully CORBA-compliant software interoperats by using the IIOP protocol. Remain to solve the problem of *cross references* between different database to have a fully interoperability of multi-vendor databases.

The grid data handling system can take advantage of the work done on *multi-agent* technology used for integrating database. In this technology the interaction with the database is done via agents namely: the client agent and the database agent, Figure 5 shows an overview of the system for querying molecular biology databases from [11]. The database agent is an abstraction of an expert on biological data domain and the user agent is an entity that performs operation on behalf of the user. The integration and database cross referencing requires a extended knowledge on the application domain, it will be rather difficult to think about a tool that automatically performs these tasks, it will be more appropriate to think of a tool that help the domain expert, who is not forcibly database expert, to easily perform the integration and cross reference of the multi-databases.

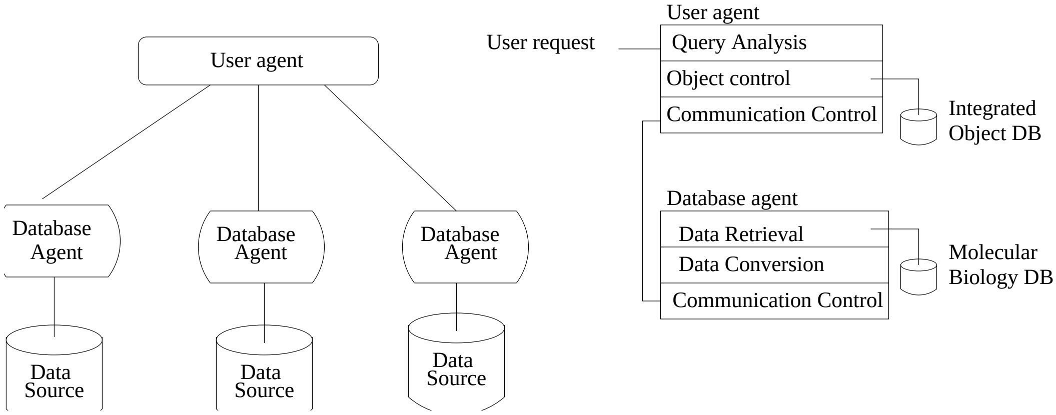

Figure 5: *Multi-agent architecture for integrating Biological databases*

6.2 Extending the GIS to support data-content manipulation

The current implementation of the GIS is not adapted to handled data-content manipulation. It is thus needed to be extended by mechanisms allowing it to to extract and/or derive from data streams the necessary metadata that allows later on grid-application to request data on the grid based on these metadata. The main taraget of the extention of the GIS can summerized in a few terms as it is diescribed in the work of Johnston et al. "dynamically increase our ability to capture, organiza, search, and provide high-performance and location independant access to large-data-objects".

The GIS needs thus to integrate a extra block that allows to integrate, coordinate, and generate metadata on the fly (Figure 6). When automating function such as data-cataloguing one key issue is the identification of units so that the process of creating the metadata can starts. Johnston et a. propose the use of either explicit marker to identify the start and the end of the data unit or implicit marker such as the time or the quantity [8].

A generic model for the metadata need to be introduced, it needs to be scalable, easy to search, and to update. A number of standard elements have proposed for the Lage-Data-object model (LDO):

- identifying information (unique id, owner, ...)
- links to the original data
- acces control fields

A class defined in the LDO model is provided with:

- search methods
- data component access control

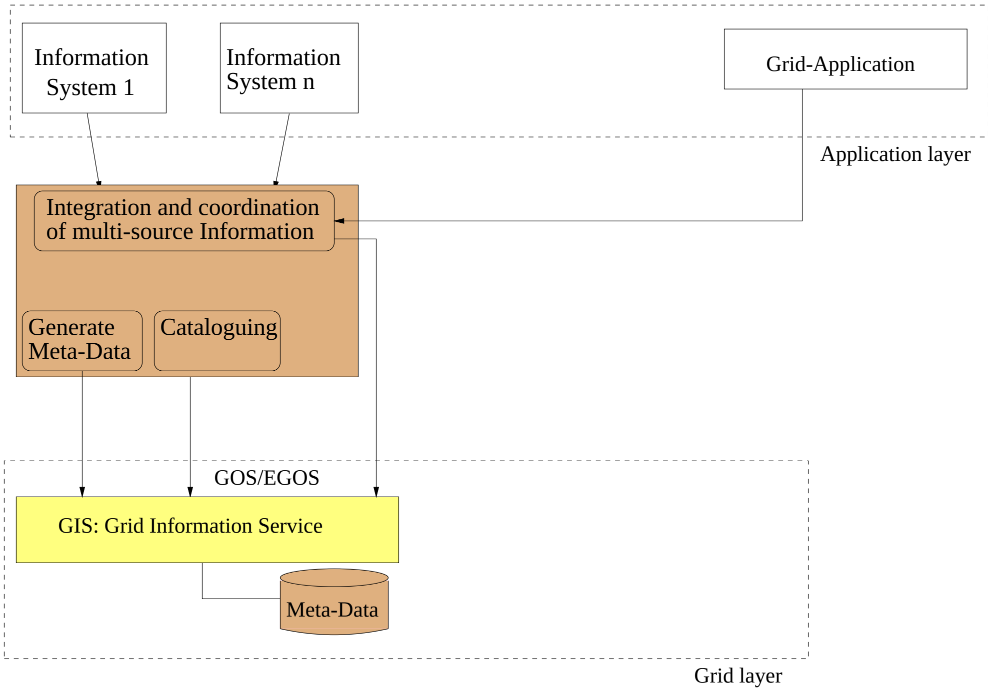

Figure 6: *Generating metadata on the Fly will bridge the Application and grid domains*

- data entry methods
- access control methods

The LDO model has been already used for a number of application to create digital library-like collectins automatically from high data-rate, real-time output e.g., satellite imaging systems, scientific instruments, health care imaging. It might be thus a good starting point to build up the GIS extension.

7 Conclusions

In this short report, we tried to point out some of the research and implementation issues related to the integration of the VLAM-G project with the grid in general and the Data-Grid in particular.

The VLAM-G project has not only to integrate with the grid technology, it has to aim at addressing some of the open issues like modeling and representing information contained in data sets manipulated within the grid. Using database technology seems a logical approach to this problem, but still a few questions remain to be answered with regard to the metadata directory services.

As was mentioned in the Sections ??, it is very important for this work to do an extensive overview of the literature in this domain to deepen our knowledge on grid technology.

The task force needed for a good and rapid progress of this work require a background on: *Grid Technology, Database Technology, and Globus Technology.*

References

[1] A. A.Goodrum. Image inforamtion retrieval an overview of current research. *Information Science*, 3(2):63–67, 2000.

[2] M. Baldonado, C.-C. K. Chang, and L. Gravano. Metadata for digital libraries: Architecture and design rationale. Technical Report Number SIDL-WP-1997-0055, Stanford University, 1997.

[3] C. Baru, R. Moore, A. Rajasekar, and M. Wan. The sdsc storage resource broker. In *Proc. CASCON'98*, Toronto, Canada, December 1998.

[4] J. Bester, I. Foster, C. Kesselman, J. Tedesco, and S. Tuecke. GASS: A data movement and access service for wide area computing systems. In *Proc. IOPADS'99*. ACM-P, 1999.

[5] A. Chervenak, I. Foster, C. Kesselman, and C. S. S. Tuecke. The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets. In *Proceedings of the NetStore*, page Online Proceeding at http://dsi.internet2.edu/netstore99/, Seattle, WA, October 1999.

[6] B. Didier, K. Schuchardt, G. von Laszewski, and S. Fitzgerald. Representing people for the grid information sevices. Proposal for the Grid forum Draft, Grid Forum, 2000.

[7] C. R. Garita. A survey of dsitrbuted information management approaches for virtual entreprise infrastructure. Chapitre of ph.d thesis, University of Amsterdam, 2000.

[8] W. Johnston, J. Guojuin, C. Larsen, J. Lee, G. Hoo, M. Thompson, and B. Tierney. Real-tome digital libraries based on widely distributed high perfornance managenent of large data objects. Technical Report LBNL-39613 UC-405, LBNL, 1997.

[9] E. Kaletas, A. Peddemors, and H. Afsarmanash. Archipel cooperative islands of information. Technical Report XXXX, University of Amsterdam, 2000.

[10] K. Marzullo, M. Ogg, A. Ricciardi, A. Amoroso, F. Andrew, Calkins, and E. Rothfus. Nile: Wide-area computing for high energy physics. In *Proc. SIGOPS'96 European Workshop*. http://mosquitonet.stanford.edu/sigops96/papers/index.html, 1996.

[11] H. Matsuda, T. I. abd Michio Nakanishi, and A. Hashimito. Querying molecular biology databases by integration using multiagents. *IEICE Trans. INF& SYST.*, E82-D(1):199–207, January 1999.

[12] R. Moore. Grid forum data access working group. Draft http://www.sdsc.edu/GridForum/RemoteData/Papers/, Grid Forum, 2000.

[13] J. Newton. Application of metadata standards. In *Proceedings of the IEEE Metadata conference*, Maryland, USA, April 1996.

[14] B. Plale and K. Schwan. dquob: Managing large data flows uding dynamic embedded queries. In *Proceedings of High Performance Distributed Computing 2000*, page XXXX, XXXX, 2000.

[15] J. Qin. Representation abd organization of information in the wev space: From marc ot xml. *Information Science*, 3(2):84–87, 2000.

[16] P. Raj and N. Ishii. Interoperability of biological databases by corba. In *Proceedings of the Inter. Conf. on Information Intelligence and Systems*, pages 16–23, Bethesda MD, October 1999.

[17] C. Salisbury. Remote data access architecture. Draft http://www.sdsc.edu/GridForum/RemoteData/Papers/, Grid Forum Data Acces Working Group, 2000.

[18] D. Tafti, W. Huang, P. Baker, R. Heiland, M. Livny, T. Tannenbaum, and S. Martin. High quality rendering of temporal field data using ncsa visbench and condor-globus gate way. In *Proc. supercomputing'99*. http://www.ncsa.uiuc.edu/~heiland/sc99/sld001.htm, 1999.

[19] P. Vanderbilt. Gxd verses direct xml. Draft (March 2000) http://www.nas.nasa.gov/~pv/gxd/, NASA Ame, 2000.

[20] G. von Laszewski. Defining schemas for the grid information services. Proposal for the Grid forum Draft, Grid Forum, 2000.

[21] G. von Laszewski and P. Lane. Mdsml: An xml binding to the grid object specificaction. Proposal Draft, Grid Forum, 2000.

[22] M. Wan, A. Rajasekar, and R. Moore. Storage resource broker data grid. Proposal -Draft- http://www.sdsc.edu/GridForum/RemoteData/Papers/, Grid Forum Data Acces Working Group, 2000.