# VLAM-G: A Grid-Based Virtual Laboratory

H. Afsarmanesh[1], R. Belleman[1], A. S. Z. Belloum[1], A. Benabdelkader[1], J. F. J. van den Brand[2,3], T. M. Breit[4], H. Bussemaker[4], G. B. Eijkel[5], A. Frenkel[1], C. Garita[1], D. L. Groep[2], A. W. van Halderen[1], R. M. A. Heeren[5], Z. W. Hendrikse[1], J. Kaandorp[1], E. C. Kaletas[1], V. Klos[6,3], P. Sloot[1], R. D. Vis[2,3], A. Visser[1], H. H. Yakali[1], L. O. Hertzberger[1]

[1]*University of Amsterdam Computer Science Department, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands*

[2]*NIKHEF, P.O. Box 41882, 1009 DB Amsterdam, The Netherlands*

[3]*Vrije Universiteit, de Boelelaan 1081, 1081 HV Amsterdam, The Netherlands*

[4]*University of Amsterdam, Faculty of Sciences, Kruislaan 318, 1098 SJ Amsterdam, The Netherlands*

[5]*FOM Institute AMOLF, Kruislaan 407, 1098 SJ Amsterdam, The Netherlands*

[6]*IP Globalnet Nederland BV, Soesterberg, The Netherlands*

**Abstract**

VLAM-G, the Grid-based Virtual Laboratory AMsterdam, provides a science portal for distributed analysis in applied scientific research. It offers scientists the possibility to carry out their experiments in a familiar environment, where content and data are clearly separated. Emphasis is put on the development and use of open standards and seamless integration of external devices.

In this paper the proposed design and prototype implementation of the VLAM-G platform are presented. In the testbed, we have applied several recent technologies such as the Globus toolkit, enhanced database systems, and visualization and simulation techniques. Moreover, several domain specific case studies are described in some detail.

# I. Introduction

Recent achievements in network technology have made it possible to make the next step in distributed computing: the Grid [1]. Nowadays, an impressive number of workstations and super computers can be connected in an efficient way, offering users unprecedented computational power. The part of the scientific community that relies on high-performance and/or high-throughput computing machinery needs an ever growing amount of CPU-cycles. Moreover, huge amounts of data are generated from experiments conducted world-wide. It would be desirable for scientists to be able to access these data remotely, thereby facilitating collaboration. Furthermore, in such a comprehensive experimentation environment access to external devices is highly preferrable.

VLAM-G offers such a distributed analysis platform for applied experimental science. It incorporates and integrates the most recent developments in Grid-computing (such as the Globus toolkit [2]), database technology and visualization techniques.

A simple classification of potential users of the Grid has been proposed in [3], where three categories have been indicated:

- the novice scientist,
- the senior scientist, and
- the application developer.

Taking the aforementioned categories into account, it is of great importance to pay particular attention to the HCI (Human Computer Interface), besides other important issues such as collaboration and security. Potential users of Grid technology are not supposed to know about all the computer science specifics involved to realize such an environment themselves. They are also not inclined to deal with the day-to-day operations concerning *e.g.*, security, resource management and networking.

Also, it is essential that this platform is designed in such a way that it can easily be adapted to include either a new application domain or a different group of users. A modular architecture is essential to establish such a flexibility. Supporting inter-disciplinary interactions implies the composition of (software) modules, which may have been developed independently [4], [5]. VLAM-G takes this modularity into account by providing a modular data flow-based system. In order to ensure that only proper connections can be established, the constituting modules communicate using a strongly typed communication mechanism.

Furthermore, domain specific application case studies are currently developed in physics, bio-informatics, and systems engineering. Bearing these application domains in mind, three different science portals are being implemented simultaneously, based on the same technology.

In addition to meeting these domain specific challenges, a common Run-Time System (RTS) has been designed which can properly deal with a set of generic modules that can be used in each of the three science portals. A preliminary requirement analysis performed together with various scientists from the three application domains has led to a classification of three types of modules:

*Control modules* : these modules are being developed to control external devices.

*Processing modules* : these modules are data filters and visualization modules.

*Database and mass storage modules* : these modules allow the experimenter to store and retrieve local and remote data either in raw data files or in databases.

A scientist assembles these modules to compose his experiment(s). VLAM-G middleware assists the scientist by providing a Graphical User Interface (GUI) and an assistant. These components ease the handling of remote data access, resource allocation, security issues and access to external devices. The editing phase of an experiment is performed using an appropriate science portal, implemented as a drag-and-drop GUI (Figure 1).

A first prototype will integrate various filters developed for physics experiments, modules for accessing high performance storage (HPSS) systems and databases, modules for specific external device control and visualization related modules.

## II. The VLAM-G science portals

Application from three different scientific domains have been selected as case studies to steer the development of the VLAM-G platform. One application is meant for the chemistry and physics domain, dedicated to handling of device-generated data within a distributed environment. These data originate from apparatus at geographically different locations and should subsequently be combined to create a virtual Surface Analysis Laboratory for materials analysis at the micrometer scale.

A second application focuses on complex design problems (*i.e.* in traffic control and telematics), and will support the evaluation of complex design by means of simulation and experiment. Bioinformatics and medicine are the prime interest of the third application domain. It will focus on integration of large databases and on simulation and modelling for medical studies. Below, these application domains are described, as well as advanced visualization and simulation techniques.

### *A. A virtual material analysis laboratory*

The use of advanced physical methods to study the properties of material surfaces has proven to be a powerful tool, not only for fundamental but also for applied research. In fields as diverse as art conservation, cancer therapy and cosmology, sub-micron techniques for elemental and molecular analysis lead to better understanding of the processes involved. These methods will henceforth referred to as “Materials Analysis of Complex Surfaces”, MACS.

The MACS centre comprises three large devices: an imaging infra-red Fourier-transform spectrometer (FTIR [6]), a 4 MeV ion accelerator (Nuclear Microprobe NMP) for elemental analysis using PIXE (particle induced X-ray emission) techniques and characteristic nuclear reactions [7], and an imaging mass spectrometer (TOF-SIMS Time-Of-Flight Secondary Ion Mass Spectrometry [REF:FIXME]). These devices largely produce complementary data. FTIR imaging is primarily sensitive to the chemical composition of materials, owing to specific molecular resonances in the infra-red domain. PIXE and specific nuclear reactions exploit the selectivity of the response of atomic nuclei to incident ions to identify the elemental composition of the sample. In addition, PIXE and nuclear reactions are able to provide precise quantitative information at the femtogram level. Existence and spatial distribution of (large) macromolecules and polymers may be studied with imaging mass spectrometry. Correlation of these complementary data would significantly add to the scientific merit of these experiments.

VLAM-G can provide experiments with a uniform environment for data analysis, since most if not all experiments share a similar process flow (in this particular case preparation of a sample, surface analysis, visual inspection and possibly re-analysis). A visual representation of such a process flow [8] is provided by the front-end.

Those parts of the process flow that represent the experiments themselves, may contain any of the above mentioned devices. Data originating from the FTIR and TOF-SIMS devices is transferred to the VLAM-G by explicit human intervention (by storing the data files in a network-connected repository). The NMP is controlled by the end-user and experimental data are generated real-time.

The subsequent data analysis may be modelled as a directed data-flow, that is suitable to be processed by the VLAM-G RTS. Data is acquired by a dedicated system connected to the data acquisition hardware. This acquisition in turn yields a continuous flow of discrete events (NMP, TOF-SIMS) or a snapshot image (FTIR). All data are converted to a common format: $n$-dimensional data 'cubes' stored as NetCDF 3 binary files following locally established conventions. These NetCDF data cubes hold a copy of part of the meta-data to enable stand-alone interpretation of these data sets outside of the VLAM-G context, *e.g.*, by other laboratories.

During the analysis of an experiment, components from the VLAM-G RTS convert and visualize these data cubes. A simplified version of such a data flow is shown in Figure 3. Generic visualization modules are provided by experts in this field (refer to Section II-D).

### *B. Simulation and systems engineering*

Interest and support for ICT simulations related to traffic are steadily increasing. The main reason is that such simulations will play an essential role in evaluating the different aspects of the communication systems involved, such as traffic management strategies and, most importantly, telematics applications.

Field tests provide an opportunity to measure the benefits of new applications in real-world environments and convey vital information for system design and evaluation. However, the scale and the range of the conditions for which the system performance can be tested is severely limited *in vivo*. Using simulation, both key operations and performance issues may be investigated under a wide range of conditions. Therefore, simulations are invaluable for a sound system design. The simulation capabilities of VLAM-G can be applied to support the evaluation of these telematics applications.

In addition to simulation, research on traffic management and subsequent analysis generate data, which will be accessed from within the VLAM-G environment, since it is especially targeted toward collection and analysis of large data sets. In fact, it has already been applied successfully in the Dutch project “Rekeningrijden” (Electronic Toll Fee Collection via Dedicated Short Range Communication). During the system evaluation tests, large amounts of traffic data have been obtained by different Electronic Fee Collection systems. The environment includes tools to parse these data from files and to store them into a database. In order to provide a comprehensive environment for data analysis, an interface between the database and Matlab has been developed in addition to generic traffic analysis and validation tools[9], [10]. Depending on any new requirements with respect to the analysis procedure, corresponding maintenance for the database is being provided.

Ongoing developments within the traffic management will bridge the gap between the VLAM-G and the Matlab environment. Once completed, users will be able to start their VLAM-G applications from Matlab and get their simulation/experimentation results directly back into their Matlab environment, where they may continue their data analysis.

### *C. VLAM-G and bio-informatics*

DNA micro-arrays allow genome-wide monitoring of changes in gene expressions. These changes may be induced by response to physiological stimuli, toxic compounds, disease or infection. Depending on the organism and the exact design of the experiment, the number of useful data points produced per experiment will range from around 120 000 for a simple organism like yeast upwards to 700 000 for complex organism such as humans.

Micro-array technology enables the study of the characteristics of thousands of genes in a single experiment:

- Identification of genes responsible for a given physiological response.
- Monitoring of physiological changes that occur during the progression of a disease or in industrial organisms to identify cellular responses at specific stages of a production process.
- Better understanding of the mechanisms of gene regulation and identification of the transcription factors responsible for coordinate expression of genes that display similar responses.
- Assignment of functions to novel genes.

A gene expression database called EXPRESSIVE has been developed to support the information management requirements of the DNA micro-array case study. In order to fully support these requirements, an extensive study of the micro-array experiments has been performed. During this study, laboratory activities (such as clone preparation, micro-array production, probe preparation, hybridization, imaging, and scanning) as well as the data analysis process have been analyzed. Both activities require information extraction and data integration functionality. A survey of both existing gene expression databases as well as ongoing standardization efforts has been carried out. As a result, a system capable of representing any experimental information generated is developed within VLAM-G.

EXPRESSIVE is capable to represent any kind of micro-array expression pattern by applying the existing and emerging standards such as the minimum information set and the Dublin Core Meta Data standard [11]. With the experience gained from the evolution of the sequence data repositories, the EXPRESSIVE design is scalable and open for extensions to support future requirements, *e.g.* to other genome databases. Summarizing, EXPRESSIVE provides:

- Definition and management of information to support the storage and retrieval of the steps and information involved in DNA microarray experiments, to support investigation and reproducibility of experiments using the VLAM-G.
- Storage and retrieval of results from DNA microarray experiments (both raw and processed) using the VLAM-G, to facilitate both analysis and scientific collaboration.

The EXPRESSIVE system elaborates on the functionality provided by VLAM-G, including the advanced federated information management functionality of ARCHIPEL. The latter provides the generic federated information management framework for VLAM-G [12]. Application of ARCHIPEL to the biology domain implies an extension of the EXPRESSIVE system targeted towards integration with to other information recources. Here one may think of databases specialized in gene expression and gene sequences, a LIMS (Laboratory Information Management System) or hospital information systems.

### *D. Visualisation and interactive simulation*

In many of the scientific computing problems that will be performed, the complexity of both experiment and generated data is too vast to analyse analytically or numerically. For these situations, “exploration environments” provide an essential means to present and explore the data in a way that allows a researcher to comprehend the information it contains. Exploration environments combine visualization and interaction functions into one system to allow exploration of large data spaces. These data spaces may originate from data acquisition devices (such as the aforementioned DNA micro-array scanners, spectroscopes, medical MRI or CT scanners, etc.) or represent results from prior processing modules (such as computer simulations).

In static exploration environments (SEEs), the data presented to the user is time invariant; once the data is loaded into the environment, the user is presented with a visual representation of this data. Interaction methods are provided to change the visualization parameters interactively in order to get the best view in order to gain understanding.

If interaction with other processing modules is also allowed, however, we speak of dynamic exploration environments (DEEs). DEEs extend the static model in that the information provided to the user is regenerated periodically by an external process. The environment is expected to provide

- a reliable and consistent representation of the results of the simulation *at that moment* and

- mechanisms enabling the user to change parameters of the external module.

DEEs have additional obligations over SEEs, such as user interaction with other modules and time management which deals with the proper exchange of time stamped information between components [13]. Given these considerations, we have chosen for the "High Level Architecture" [14] as a suitable architecture for constructing a DEE. The High Level Architecture (HLA) aims to establish a common architecture for simulation to facilitate interoperability among simulations and promote the reuse of simulations and their components.

An exploration environment provides the opportunity to interact with a living simulation. This interaction can take any form; from typed input for simple types of interaction via GUIs to fully immersive virtual environments. The main feature of immersive environments (such as a CAVE) over other GUIs is that user-centered stereoscopic images are presented to a user rather than visualization centered three-dimensional (3D) projections and that direct interaction with these presentations is possible.

For coordination between the different components in a DEE we use Intelligent Agents (IAs). IAs are software modules capable of performing three functions:

1. perceiving state changes in the environment through the use of monitors,
2. taking actions that affect conditions in the environment and
3. reasoning to interpret perceptions, solve problems, draw inference, and determine actions. Agents execute autonomously, interfering minimally with the rest of the environment, apart from communicating with other agents or the user.

Currently we use agents to provide feedback to the user concerning the state of a simulation (*e.g.* accuracy of the simulation, time to completion, convergence rate), user interaction (including speech recognition and synthesis) and context sensitive assistance.

As a case study to validate the concepts described above, we apply this system to a medical application: simulated abdominal vascular reconstruction. This test case contains all aspects of a DEE that are of consequence in the construction of a generic environment. Our aim is to provide a researcher with an environment in which one can try out a number of vascular reconstruction procedures, using patient specific angiographic data, and see the influence of these using flow simulation.

## III. Architecture of VLAM-G

One of the fundamental challenges in experimental science is the extraction of useful information from large quantities of data. Another characteristic, nowadays gaining attention, is the need for cooperation of multidisciplinary teams spread over various geographically distinct locations. VLAM-G meets these challenges:

- the VLAM-G as an application environment, for which new technology will be developed, and
- the VLAM-G as GridWare, developing generic (software) tools to serve as a basis for application environments.

The above mentioned science portals are build with the application enviroment in mind.

Grid technology is meant to provide services to facilitatie interoperability of heterogeneous systems [2]. The VLAM-G project elaborates on this new technology, as was explained in detail in [15]. The VLAM-G design promotes the use of components from the Globus toolkit, which is the *de facto* standard in Grid computing nowadays. In fact, most of the middleware is based on existing Globus tools, thereby possibly profiting of future developments within the Globus project [1].

Within the VLAM-G Gridware work-package, two activities can be distinguished. The VLAM-G Abstract Machine provides an integration toolbox, connecting application environments to an existing Grid infrastructure.

VLAM-G is based on a parallel/distributed hardware environment, connected by a local Gigabit Ethernet LAN. During development, this infrastructure will be extended to a WAN on national scale already.

### A. *The VLAM-G Architecture*

The VLAM-G fits a modular 4-tier architecture, see Figure 4. The four tiers of the VLAM-G consist of the application tier, the application toolkits tier (VLAM-G science portals and the RTS), the grid middleware tier (providing the grid services) and the grid fabric tier (for the access to underlying resources).

In order to store the information relevant to a scientist, a generic database model has been developed, with specific instances for each application domain separately. A study of the application domains driving the VLAM-G development has allowed us to identify generic aspects valid for these different domains. By combining generic features that characterize scientific experiments with those of specific case-studies, it is possible to develop a database model for the Experimentation Environment (EE). The EE data model [16] focuses on storage and retrieval of information to support a generic definition of steps and information representing a scientific experiment, in order to investigate and reproduce experiments. The EE data model allows a random ordering of processes and data elements, which enables one to construct an arbitrary process data flow. The EE data model has already been applied to build the information management system for the MACS [8] and the EXPRESSIVE [11] application databases.

Another consequence of the case studies is its implication for the data and process flow for each Grid application. Together with the processes which relate to data it extends the EE data model adding domain specific features.

Experiments are embedded in the context of a *study*. A study is defined by a series of steps intended to solve a particular problem (in an application domain) which boils down to a process flow. As such, a study generally comprises steps performed using the VLAM-G Run-Time System, process steps performed outside the scope of the RTS (*e.g.* manually, and the description of data elements related to objects or process steps.

A *process flow* is the result of a commonly agreed work flow in a specific application domain, *e.g.* space-

[1] http://www/globus.org

resolved surface analysis or genome expression studies. Such process flows have been developed for chemo-physical analysis of samples using space-resolved techniques (material analysis of complex surfaces) and for biogenetic experiments.

A PFT (process flow template) is used to represent such a workflow. The user is guided through this process flow using context-sensitive interaction. Each PFT has a starting point, for instance the source object for the sample under study. Its characteristics (meta data) should be supplied by end user, *e.g.*, by a fill-in form. Once the 'source' data are supplied, related items become selectable. Data elements within the PFT require object meta data to be supplied, process steps require a description of the procedure applied to particular objects. In the case of a process which is executed by the VLAM-G RTS, selection will initiate a VLAM-G experiment. Associated meta data are registered (using a fill-in-form) and supplied implicitly.

A VLAM-G RTS experiment consists of process steps, input and output data objects and their topology. Note that meta data are stored in the application database, whereas any resulting 'base data sets' maybe stored outside of this context, depending on the application. Only references to these sets are retained.

### *B. The VLAM-G Abstract-Machine*

The VLAM-G Abstract-Machine (VLAM-G-AM) is middleware, that enables the end-user to run experiments within the VLAM-G environment. As such, it acts as an intermediate layer between the Grid-layer (Globus) and the VLAM-G users, hiding details on Grid technology which are irrelevant to the users carrying out experiments.

The VLAM-G-AM provides services such as editing , scheduling, running, monitoring, and publishing results of scientific experiments. These experiements are defined by selecting a number of processing elements (also referred to as modules) and are connected in a such a way that their topology reflects the data flow of a particular experiment. Modules [2] are independent of each other and have generally been developed by different persons. Therefore, the VLAM-G-AM has to offer:

- a transparent layer which communicates data between these processing elements and
- a platform enabling the modules to execute their data processing and internal computations.

In terms of a multi-tier architecture, the VLAM-G-AM is just another specialized piece of middleware, which enables the end-user to run VLAM-G specific experiments. Let us for the moment assume a generic 4-tier architecture, then the VLAM-G-AM can be positioned in the third tier, as indicated in Figure 5.

The Abstract Machine offers seamless access the software and hardware resources of VLAM-G. The information needed to provide such a service will be offered by meta data stored either in a database or in the Globus Meta-computing Directory (MDS). A connecting layer contains interfaces to the subsystems managing the meta-data directories:

[2] Separate modules may be grouped together, to form a larger so-called super-module.

- interface querying the database,
- interface transmitting user requests to the global computing resources.

The VLAM-G-AM extends Grid discovery features provided by the Globus Toolkit, since it allows for retrieval of information related to previous experiments. In addition, it offers VLAM-G specific data on processing modules and experiment topologies.

## IV. Conclusions

We have presented the design of a high performance-distributed computing platform for experimental science. The VLAM-G offers scientist easy access to Grid resources, both in software and hardware. The application of this concept has been shown to work very well for three separate scientific domains: bio-informatics, systems engineering and physics.

It has also been shown that the VLAM-G promotes reuse of existing components:

- It elaborates on existing (Globus) Grid technology
- It allows scientists from different disciplines to reuse generic data flow modules

.

Particular attention has been paid to construct a user friendly environment, using either context sensitive information, automated resource management and assistance during the design process. The latter is established by means of an assistant, which helps users during the design process by offering them pre-defined process flows for typical experiments.

## Acknowledgments

This work was supported by an ICES/KIS grant from the Dutch ministry of education and sciences (OC&W) and the ministry of economic affairs (EZ). It is also part of the research programme of the Foundation for Fundamental Research of Matter (FOM), which is financially supported by the Netherlands Organization for Scientific Research (NWO). This project includes software developed by and/or derived from the Globus project (`http://www.globus.org/`).

## REFERENCES


[1] I. Foster and C. Kesselman. Realtime widely distributed instrumentation system. In *The Grid: Blueprint for a Future Computing Infrastructure*, pages 83–103. MORGAN-KAUFMANN, 1998.

[2] I. Foster and C. Kesselman. The Globus project: A status report. In *Proceedings of the Heterogeneous Computing Workshop*, pages 4–18. IEEE-P, 1998.

[3] Gregor Von Laszewski, Ian Foster, Jarek Gawor, Peter Lane, and Mike Russell. Designing grid-based problem solving environments and portals. white paper, Argonne National Laboratory, 2000.

[4] A. Abd-Allah and B. Boehm. Reasoning about the composition of heterogeneous architecture. Technical Report UCS-CSE-95-503, University of Southern California, 1995.

[5] David Garlan, Robert Allen, and John Ockerbloom. Architectural mismatch: Why reuse is so hard. *IEEE software*, Vol. 12(No. 6), November 1995.

[6] G.B. Eijkel, H. Afsarmanesh, D. Groep, A. Frenkel, and R.M.A. Heeren. Mass spectrometry in the amsterdam virtual laboratory: development of a high-performance platform for meta-data analysis. In *Proceedings of the 13th Sanibel Conference on Mass Spectrometry: informatics and mass spectrometry*, Sanibel Island, Florida (USA), January 2001.

[7] R.D. Vis, J.L.A.M. Kramer, G.H.J. Tros, F. van Langevelde, and L. Mars. The upgraded amsterdam nuclear microprobe. In *Proceedings of 3rd Int. Conf. on Microprobes*, Uppsala, Sweden, 1992.

[8] A. Frenkel, H. Afsarmanash, G.B. Eijkel, and L.O. Hertzberger. Information management for physics applications in the virtual laboratory environment. Technical Report CS-2001-03, University of Amsterdam, 2000.

[9] A. Visser, A.J. van der Wees, and L.O. Hertzberger. Discrete event modelling methodology for intelligent transport systems. In *Proceedings of the World congress on Intelligent Transport Systems*, page paper 2016, Torino, Italy, 2000.

[10] A. Visser, H.H. Yakali, A.J. van der Wees, M. Oud, G.A. van der Spek, and L.O. Hertzberger. An hierarchical view on modelling the reliability of a dsrc-link for etc applications. Technical Report (submitted to IEEE Transactions on Intelligent Transportation Systems) CS-99-02, University of Amsterdam, 1999.

[11] E.C. Kaletas H. Afsarmanesh, T.M. Breit, and L.O. Hertzberger. Expressive - a database for micro-array gene expression sutdies. Technical Report CS-2001-02, University of Amsterdam, 2000.

[12] E.C. Kaletas, A. J. H. Peddemors, and H. Afsarmanesh. Archipel - preliminary analysis of requirements. Technical Report CS-2001-06, University of Amsterdam, 2001.

[13] R.G. Belleman and P.M.A. Sloot. The design of dynamic exploration environments for computational steering simulations. In *Proceedings of the SGI Users' Conference 2000 (SGI'2000)*, Krakow, Poland, October 2000.

[14] R.J. Allen, David Garlan, and James Ivers. Formal modeling and analysis of the HLA component integration standard. In *Proceedings of the Sixth International Symposium on the Foundations of Software Engineering (FSE-6)*, November 1998.

[15] A. Belloum, Z.W. Hendrikse, D.L. Groep, E.C. Kaletas, A.W. van Halderen, H. Afsarmanesh, and L.O. Hertzberger. The VLAM-abstract-machine: a data & process handling system on the grid. In *to be published at the HPCN2001*, Amsterdam, The Netherlands, June 2001.

[16] E.C. Kaletas, H. Afsarmanesh, and L.O. Hertzberger. Virtual laboratory experimentation environment data model. Technical Report CS-2001-01, University of Amsterdam, 2001.

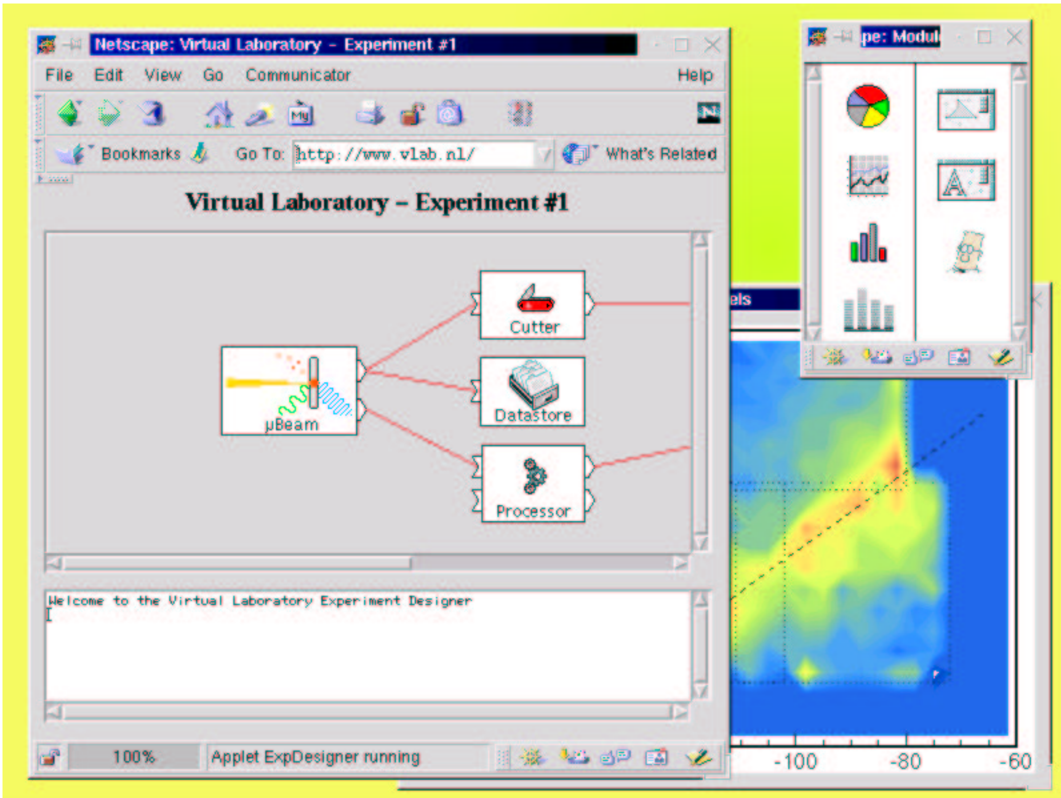


Fig. 1. VLAM-G main interface for the definition of an experiment

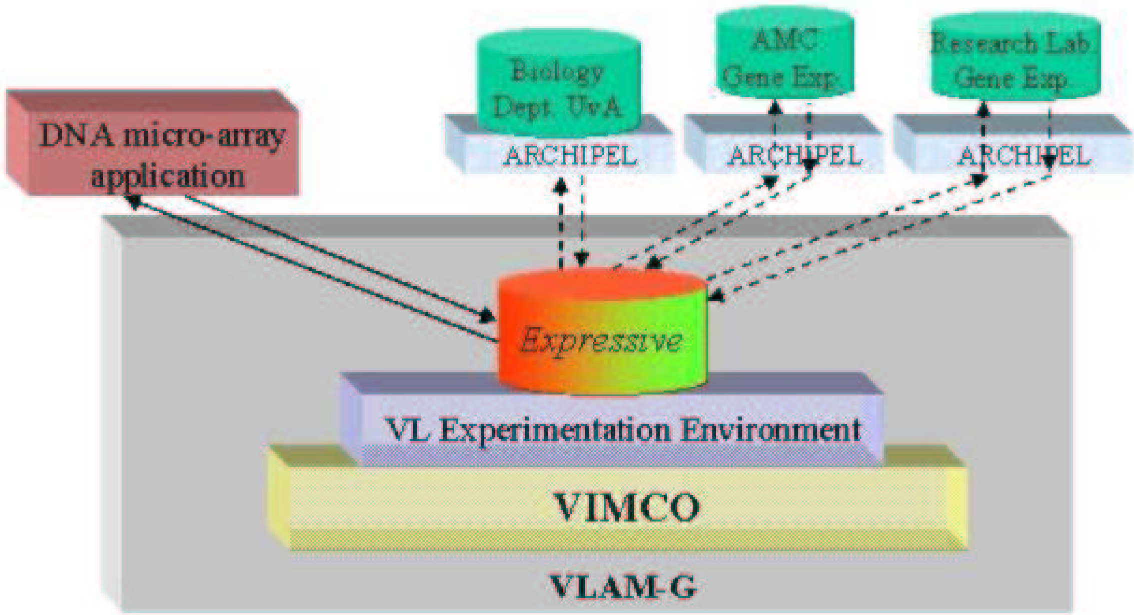


Fig. 2. Overview of EXPRESSIVE system

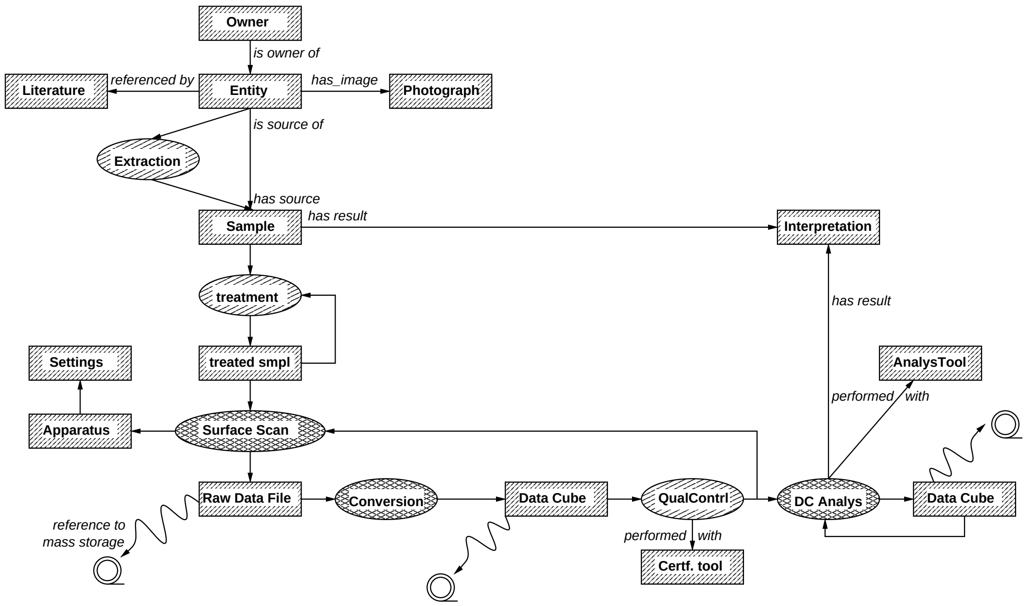


Fig. 3. Typical steps in a surface-analysis study, as represented by the MACS [8] process flow. The hatched rectangles represent meta data associated with objects (either physical objects or bulk data on mass storage). Hatched ovals represent operations: cross-hatched ovals are performed using the VL-AM Run Time System, the single-hatched ovals represent meta data associated with manually performed process steps.

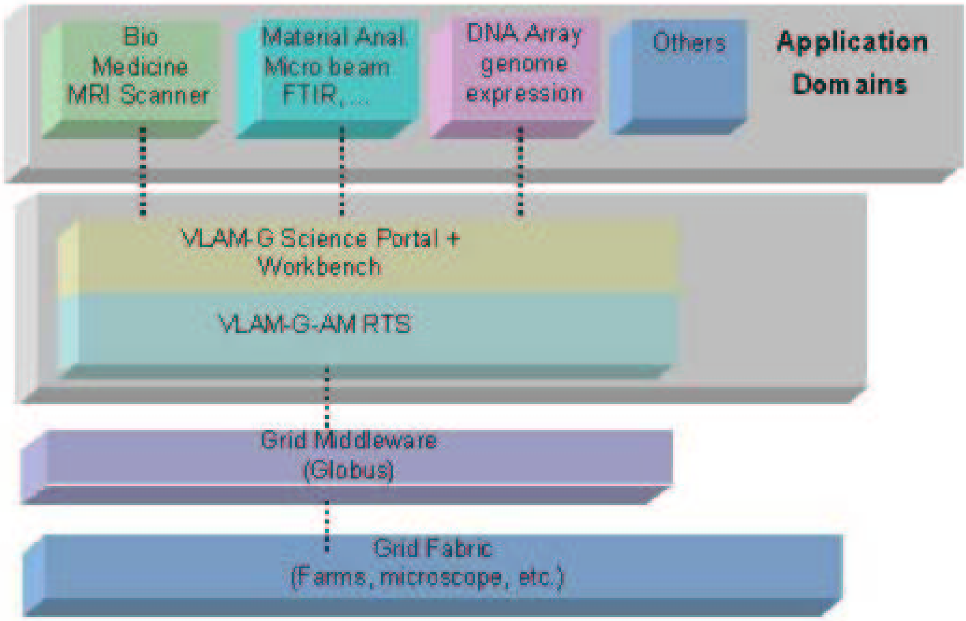


Fig. 4. Overview of VLAM-G Architecture

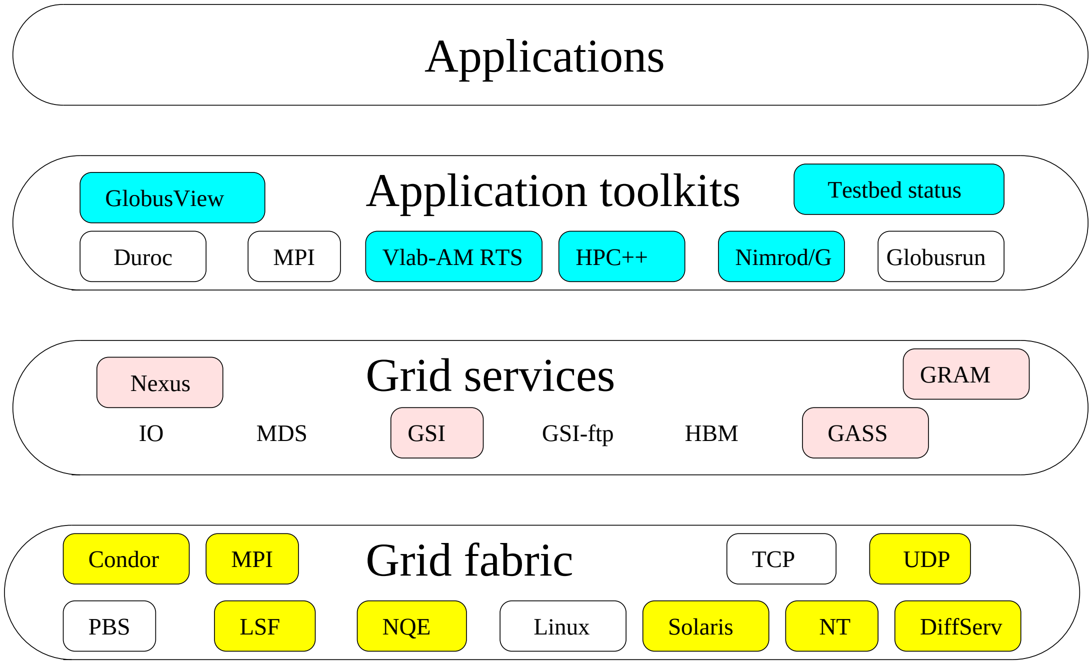


Fig. 5. Schematic overview of the position of the VLAM-G-AM in the commonly agreed on 4-tier Globus/Grid architecture.