A reference architecture for scientific virtual laboratories

H. Afsarmanesh, E.C. Kaletas, A. Benabdellkader, C. Garita, L.O. Hertzberger

Department of Computer Science, Computer Architecture and Parallel Systems Group,
University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands

Abstract

Recent advances in the IT can be applied to properly support certain complex requirements in the scientific and engineering domains. In experimental sciences, for instance, researchers should be assisted with conducting their complex scientific experimentation and supporting their collaboration with other scientists. The main requirements identified in such domains include the management of large data sets, distributed collaboration support, and high-performance issues, among others. The virtual laboratory project initiated at the University of Amsterdam aims at the development of a hardware and software reference architecture, and an open, flexible, and configurable laboratory framework to enable scientists and engineers with working on their experimentation problems, while making optimum use of modern information technology approaches. This paper describes the current stage of design of a reference architecture for this scientific virtual laboratory, and focuses further on the cooperative information management component of this architecture, and exemplifying its application to experimentation domain of biology. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Virtual laboratory; Digital experimentation environment; Collaboration; Information management

1. Introduction

The problems which scientists and engineers face when designing and developing advanced applications in such domains are quite diverse and complex in nature. The complexity is faced, for instance, when it is necessary to remotely control/monitor a physical apparatus, run activities that require excessive computational resources, request collaboration within a distributed community that involves scientists with different interests, look for the necessary information requested by the application from various information resources, or when it is necessary to share the local results with external applications from the outside world. However, the rapid improvement in the networking technology, distributed computing systems, and federated information management methodologies allow these application developers to solve some of these problems.

Ongoing design and development work in the area of virtual laboratory (VL) is mostly focused on research on certain specific aspects, for instance, mostly related to the distance problem. Here, for instance, they introduce mechanisms to remotely control devices, for the video conferencing, and file sharing mechanisms to enable the co-working among scientists [1–3].

Depending on the widely varying application interests (e.g., education, games, experiments, chemistry, aerospace, and government) and due to the variety of the possible application fields, the concept of the “VL” has been associated to many meanings. In the

0167-739X/01/$ – see front matter © 2001 Elsevier Science B.V. All rights reserved.
PII: S0167-739X(01)00042-5
area of education, e.g., VLs can be applied to start a chemical reaction, or to see the basic mechanics rules at work while sitting in front of the computer screen [4]. In other areas, VLs are used as a simulator to study dangerous situations, for instance, in the fields of aerospace [5] and nuclear engineering [6].

Moreover, the VL projects have, in general, a limited focus on a certain specific problem. For instance, under the distributed, collaborative experiment environments (DCEE) program of the US Department of Energy [17], two inter-related sets of projects are supported, either focused on the testbeds or being the technology projects. The main focus of the program can be summarized as enabling remote operations at physical apparatus and providing a history of activities through a laboratory notebook. The projects in this program include, e.g., the LabSpace — a national electronic laboratory infrastructure, distributed computing testbed for a remote experimental environment, and the collaboratory development in the environmental and molecular sciences. In [17], detailed descriptions and references to each of these projects can be found.

As such, most of the interpretations of the VL concept and related projects are limited in the sense that they provide a solution to certain specific problem in certain specific application domain. In fact, there are only a limited number of cases where the VL is not restricted to one particular case. In [7], the need for such an open environment for VL is explained, and the main requirements for such VLs are identified. Our observation, however, indicates that such an open environment is not yet developed and the scientific community still lacks a reference framework that covers many aspects of a real collaborative multi-disciplinary experimental environment.

The 4-year VL project initiated at the University of Amsterdam aims at the design and development of an open, flexible and configurable laboratory framework providing the necessary hardware and software to enable scientists and engineers to work on their problems via experimentation, and overcome many obstacles while making optimum use of the modern information technology. The VL will develop the necessary technical and scientific computing framework to fulfill the requirements in several scientific application domains. The domains of experimentation such as physics, biology, and systems engineering are considered in specific, for which the project will develop certain application cases during the first phase of its prototyping and evaluation. On one hand, the project plans to fulfill the main requirements of these domains, as the VL generic functionalities, e.g., the support for collaboration among scientists, and management of federated information, among many others. On the other hand, the framework will be flexible and configurable, in order to be extended and to support specific application-oriented requirements. The application cases mentioned above will be used to test this flexibility of the framework. This paper describes our ideas on the design of an architecture, representing the fundamental functionalities to be supported by the laboratory.

The remainder of the paper is organized as follows: Section 2 describes the VL requirements and important considerations taken into account, when designing such a framework. The architecture for the VL is described in Section 3 called VL reference architecture. The virtual-lab information management for cooperation (VIMCO) component of VL is described in Section 4, followed by an example application case for the VIMCO component, addressing the DNA micro-array. Finally, Section 6 concludes the paper.

2. VL requirements

There are certain requirements and considerations that one has to carefully take into account when designing a generic multi-disciplinary VL framework, which are inherent in the experimental science domains. Moreover, these domains state important characteristics and requirements. Therefore, the architecture of VL must be able to satisfy the requirements imposed by these characteristics and requirements.

One of the most important characteristics of the experimental science domain is that researchers need to manipulate large data sets produced by physical devices. Converting this data into valuable information involves the application of different processes on these data sets. These processes typically demand high-performance computing support and large data storage facility. Also, the efficient utilization of the data sets is becoming mandatory to support the collaboration among the ever-increasing number of experts in the experimental science domains. Furthermore, due
to the enormous amount of generated data, there is a need that the data sets resulting from different sorts of measurements are combined and inter-linked together to provide a better insight on the problem under study. These characteristics are fundamental to the experimental science, and hence indispensable to the VL.

Considering the several application domain characteristics presented above, there are many design issues that need to be incorporated in the VL architecture. Below, we address three of these key issues for consideration, namely, the proper management of large data sets, information sharing for collaboration, and distributed resource management.

The first issue addresses the management/handling of large data sets for which the size of data generated by the devices connected to the VL can be very large. The data size ranges, for instance, between 1.2 MB for the data generated every second by a micro-beam device and 60 GB for every slice scanned from a human brain tissue, generated by a CT scanner. This excessive amount of data must be stored, filtered, classified, summarized, merged, inter-linked, and made available to the programs using it with the required performance [9].

Advanced collaboration facility is the second characteristic to take into account for the VL architecture design. Not only the efficient storage and retrieval of such data is challenging, a more challenging issue is to provide means to efficiently exchange such large amounts of data among collaborating scientists, organizations, or research centers. Moreover, the data are inherently of a heterogeneous nature and there may not even exist a standard representation for such a data. One of the primary goals of the VL is to support the increase of collaboration among scientists and scientific centers. This brings up the need for a comprehensive, advanced interoperation/collaboration facility, which must be supported by the Internet tools, Web tools and other support tools.

Furthermore, the distributed nature of the VL together with the high performance and massive computation and storage requirements brings up the third important characteristic, the distributed resource management issue. Therefore, utilization of an adequate hardware resource manager within the VL must also be considered.

3. VL reference architecture

Although the detailed description of the reference architecture designed for the VL project is outside of the scope of this paper and is the subject of a forthcoming paper, here a brief description of this architecture and its components is presented. The general design of the VL architecture is based on three main architectural components (see Fig. 1):

1. **End-user application environment.** End-user application environment contains the application cases (VL scenarios) of the target scientific domains. It supports the application-specific functionality.

2. **Middleware.** Middleware is an abstract machine together with a front-end interface to enable the VL users to define and execute their experiments. It functions as a middleware between the client side applications and the low-level distributed computing facilities.

3. **Distributed computing environment.** Distributed computing environment supports the efficient utilization of the computing and communication resources at the VL.

These architectural components and their sub-components are further briefly described below.

3.1. Distributed computing environment

The computing and networking component provides the high-bandwidth low-latency communication platform, which is necessary both for making the large
data sets available to the collaborating experts and for the physical or logical distribution of the connected external devices and the client community that uses the laboratory facilities. The gigabit networking technology being set at the University of Amsterdam, and the Globus distributed resource management system [8] are considered for the development of this distributed computing environment.

3.2. Middleware

In brief, the realization of the middleware considers a development that involves several base functional components. The three main functional components so far considered for the VL, that form the skeleton of VL middleware as shown in Fig. 1, include: VIMCO, virtual-lab information management for cooperation; ComCol, communication collaboration; ViSE, virtual simulation and exploration environment. This approach enables a clear description and division of the required generic functionality for VL. As such, every one of the mentioned functional components will identify the necessary basic operations that it needs to provide. Different operations provided by these components are later integrated through the VL integration architecture. This integration will pave the way for the VL abstract machine to be able to execute any experiment that is properly defined through the VL user interface environment. In fact, the execution of every such experiment will be transformed into the execution of a set of basic operations defined for the VL abstract machine by the three main functional components of the VL middleware.

- **VIMCO.** The VIMCO cooperative information management component provides archiving services as well as the information handling and data manipulation within the VL. This layer supports a wide range of functionality ranging from the basic storage and retrieval of information (e.g., for the raw data and processed results) to advanced requirements for intelligent information integration and federated database facilities to support the collaboration and information sharing among remote database sources and centers.

- **COMCOL.** The COMCOL component enables the communication of users with external devices connected to the laboratory, as well as the secure communication and collaboration between users, both within and outside the laboratory. The Web will play the major role for this communication.

- **ViSE.** The ViSE component provides a generic environment in which scientific visualization, interactive calculation, geometric probing, and context-sensitive simulations are supported.

3.3. End-user application environment

For application-dependent cases of the VL, necessary interfaces will be provided and certain application-specific and domain-specific tools will be developed in order to enable users to define and run their experiments, using the functionality provided by the other components in the VL.

This primary functional architecture represents the generic capabilities supported by the VL. Moreover, it represents the fact that, on one hand, different components of the VL can be developed simultaneously and somewhat independently, as a part of the general design and realization of the VL, and thus addresses the openness and extendibility of the VL architecture. On the other hand, it provides possibilities for a clear description of a set of necessary primitive laboratory operations and components and their individual functionalities. These primitives later define the set of operations that can be executed by the VL abstract machine, to be described in detail in forthcoming VL publications.

4. Information management in VL — VIMCO

The information management component of VL, namely, the VIMCO, is an environment to support the manipulation of the data within the VL platform. Considering the wide variety and excessive amount of data handled within different components of the VL, the required information management mechanisms may vary from parallel to federated systems. Furthermore, the information management system shall provide different kinds of functionality, for instance, support for structured as well as binary data access, integration of data from several sources, location transparency for remote data access, secure and authorized access to shared data among different nodes, intelligent data handling, and data mining.
The general design objectives of VIMCO within the VL framework cover the fundamental database research and development to support complex domains. For simplicity reasons, only three main focus areas of development are addressed in this paper. The first area focuses on the data archive; archiving the wide variety of data necessary to be handled within the VL, supporting their temporary storage. The second area concentrates on the development of ARCHIPEL; a generic cooperative information management framework supporting the node autonomy, and the import/export of data based on information visibility and access rights among nodes. The last area focuses on the analysis, modeling, and provision of support mechanisms for the information management requirements of a specific application, the DNA micro-array application from the domain of bio-informatics.

4.1. Data archive

The work on data archiving focuses on the design and development of an information brokerage system to archive the wide variety of data with different modalities and from different sources. This includes all the data generated through specific research and application domains supported by the VL. So far, the development of the “meta-data” for the VL archive has been covered, and the necessary data storage and manipulation functionality and facility are developed. The meta-data development is based on the Dublin Core Meta Data standard [18] and included the base archive object model. The archiving database system supports a wide variety of large structured and non-structured VL data sets, with different modalities, and from different sources. This catalog/archive schema has been refined to achieve a more scalable and extendable archive meta-schema, which is able to capture the raw/processed data, the experiment and scientist related information, and hardware (devices) and software characteristics. The designed schema is easily extendable to cope with the future modifications with the addition of new experiment types.

4.2. ARCHIPEL

The research and development performed within this generic cooperative information management framework covers the fundamental data management infrastructures and mechanisms necessary to support the forthcoming advanced applications for the VL. ARCHIPEL is an object-oriented database management system infrastructure, supporting the storage and manipulation of inter-related data objects distributed over multiple sites. ARCHIPEL supports a dynamic and wide variety of database configurations (e.g., distributed or federated) and unites different object distribution strategies in a single architectural model. The framework defined for ARCHIPEL improves the accessibility to large databases in data intensive applications and provides access to a variety of distributed sources of information. The architecture of ARCHIPEL has its roots in the PEER federated/distributed database system [15], which has been previously developed within the University of Amsterdam. PEER federated database systems and its second and third generation developments, the DIMS and the FIMS systems, are already applied to several ESPRIT projects [10–14]. In the ARCHIPEL system, the nodes store and manage their data independently, while in the case of common interest, they can work together to form a cooperation network at various ARCHIPEL levels. This nesting of nodes at different ARCHIPEL levels allows for a variety of configurations, where, for instance, certain kinds of cooperation are formed to enhance the performance of data sharing, while others are formed to enhance the complex data sharing in a distributed or federated fashion.

4.3. DNA micro-array application

In this focus area, information management requirements of the bio-informatics domain, in specific the DNA micro-array, are studied and the necessary functionality to support this domain is designed. Special emphasis is paid to the nature, structure and size of the local databases as well as the integration of data from several other information sources.

The DNA micro-array application aims at

1. Definition and management of information representing both the steps and the components involved in DNA micro-array experiments, to support the further investigation and reproducibility of these experiments through the VL.

2. Storage and retrieval of experiment results (both the raw data and processed/filtered analysis
The DNA micro-array application described in more detail in the next section.

5. Application case: the DNA micro-array

DNA micro-arrays allow genome-wide monitoring of changes in gene expressions; induced in response to physiological stimuli, toxic compounds and disease or infection [16]. Depending on the organism and the exact design of the experiment, the number of useful data points produced per experiment will range from around 12,000–20,000 for a simple organism like yeast upwards to 200,000–300,000 for man.

Today, the micro-array technology makes it possible to study the characteristics of thousands of genes in a single experiment. Thus, it enables the following tasks:

1. Identify genes responsible for a given physiological response.
2. Monitor physiological changes occurring during disease progression or in industrial organisms to identify cellular responses at specific stages of the production process.
3. Better understand the mechanisms of gene regulation and identify transcription factors responsible for coordinate expression of genes displaying similar responses.
4. Assign functions to novel genes.

5.1. The micro-array experiment

Within each DNA micro-array experiment, one or more experimental conditions are defined and adjusted. These parameters represent different RNA extractions from samples of various environmental conditions (e.g., temperature). As shown in Fig. 2, every micro-array experiment is mainly accomplished in three steps: pre-experiment, experiment, and results analysis.

5.1.1. Pre-experiment

This is the stage where the necessary information about the genes, properties of the RNA extractions, results of current research, and images are gathered. This information may be locally available (in the local database), or can be accessed from remote information servers.

After gathering the necessary information, the DNA pieces (genes) to be analyzed are prepared and spotted on the array (by an array preparation device called arrayer or array spotter), and the RNAs are extracted from the samples and labeled for hybridization.

5.1.2. Experiment

During the experiment stage, the array is hybridized (put into reaction) with the RNA solution(s). A scanner then scans the hybridized array. The information about the experiment, in general, and the steps involved are stored in the database. This information consists of

- information about the genes on the array, including the location on the array;
- information about the samples and the RNAs extracted from the samples;
- experimental conditions (parameters for the experiment);
- information about the experiment and steps involved (aim, date, protocols, etc.);
- information about the scientist.

5.1.3. Analysis

In this stage, the results of the experiment are received from the scanner as images of the array. These images are fed into a software program and quantified to see the effect of a change in the parameters. The generated data from the image analysis program are stored in the database.

The locally generated data are later retrieved and analyzed. This step may also involve retrieving data from remote information sources, and analyzing the whole set of data.

5.2. Data management approach

The DNA micro-array application requirements have been studied and addressed as a case of the VL project. In this paper, specific attention is dedicated to the functionality required from the VIMCO component. From the scenario described above, two main basic requirements are identified for the proper VIMCO information management of such a large inter-linked scientific data:
Fig. 2. General description of a micro-array experiment.

1. Ability to compare experiment results with both local data sets and with other results made available by external scientific centers, and ability to easily visualize the response patterns.

2. Ability to link experiment results for each gene with other local and external information on that gene. This information is extremely heterogeneous and may involve a wide range of elements being, for instance, a sequence (sequence variants, mutations, possible association with disease), the predicted protein sequence, 3D structure, cellular location, biochemical function, interactions with other molecules, or regulation pattern.

Besides these basic requirements, many advanced data management functions are necessary to organize and analyze the large-scale expression data that is generated by the micro-array experiments, such as:

- management of large quantities and wide variety of data;
- identification of patterns and relationships on individual experiment data and also across multiple experiments;
- query processing;
- data loading with data from different sources;
- integration of external information resources;
- data mining.

Furthermore, the stored results of the experiments can be made available via Internet by means of different Web applications. These applications include results/summaries browser, SQL query evaluators, and image analysis.

Related to this Web information access, there is also a need for remote data gathering support. This process corresponds to the use, interpretation, and
storage of the information that is offered by other remote systems by means of Web interface tools. At this point, the information from the other remote sources is browsed, selected, and may be partially stored in the local database or separately in another repository.

5.3. Functionality required from VIMCO

Studying the micro-array scenario enabled us to identify the functionality that VIMCO should support for the DNA micro-array application. In this section, these requirements are briefly addressed (see Fig. 3).

**VIMCO database server.** The VIMCO database server back-end is currently represented by the Oracle and the Matisse database servers.

**Local database management.** The local database management functionality basically encompasses the “traditional” database application services, which covers the tools such as data loaders, format converters to load specific data into the database, ODBC access, XML manipulation tools, and specialized functions for VIMCO to develop higher-level functions to support specific operations on the database.

**VIMCO user management.** This functionality includes user profiles management, user authentication, and security issues.

**High-performance components.** The high-performance techniques include parallelization, physical data distribution, intelligent networking and caching techniques, and functionality provided by Globus.

**Federated schema manager and query processor.** The federated database architecture applied in VIMCO is in charge of the management of data sharing, exchange, and integration from many autonomous data sources. The queries in a federated system, in general, need to be decomposed, sent to different nodes, evaluated on their proper export schemas, results sent back to the originating node and merged to make the final result.

**VIMCO Internet applications.** The coupling of the database techniques with the Web technology is required. The services/applications include the basic Web-database tools which involve the use of already existing technology such as JDBC, XML, etc. in order to provide Web access to database information.

**VL components interoperability agent.** Depending on the interoperability approach among the VL modules (e.g., based on RPC, sockets, etc.), VIMCO will develop specific functions complying with the given “internal VL standard”, in order to make its services available to other VL modules.

**Data mining tools/applications.** Once the information about the experiments is collected it must be processed, analyzed, and the results must be presented in such a way that valuable knowledge could be gained from the identification of patterns within the data.
6. Conclusions

It is foreseen that increasing computational power and using high-bandwidth network structures in the information and communication technology will play a major role in near future for the emerging complex application domains and their requirements. Collaborative scientific application domains certainly are in need of these technology advances. The VL framework described in this paper is a step in this direction providing an open, flexible environment to support both current and future applications and their emerging requirements, for the challenging field of scientific collaboration. In this framework, the paper focuses on the design of VIMCO, the cooperative information management component of the VL project at the University of Amsterdam. VIMCO aims at supporting the requirements set for the information sharing and exchange among a wide variety of collaborating scientists in the VL. The paper addresses a reference architecture for the VL development, and some details about the functionality of the VIMCO. Furthermore, one specific application case for VL, the DNA micro-array, is described and the VIMCO functionality required to support this application is addressed.

References


H. Afsarmanesh is an Assistant Professor at the Computer Science Department of the Faculty of Science of University of Amsterdam in the Netherlands. She has been involved and has directed the research in several European (ESPRIT, DUTCH-HPCN, DUTCH-SION) and American-funded projects. At the Faculty of Science, she coordinates the research and development in the area of cooperative, interoperable, and federated databases. She has served as the Program Chairperson in International Conferences and Workshops in the area of information management and expert systems.
E.C. Kaletas is a PhD student at the Computer Science Department of the Faculty of Science of University of Amsterdam in the Netherlands. He received his BSc from the Middle East Technical University in 1997. Recently, he has been involved in several national and international research projects, focusing on analysis, design and development of federated information management systems in the domains of bioinformatics, virtual laboratories (problem solving environments), and engineering. His current research is on the advanced mechanisms for information integration and privileged access to autonomous nodes.

A. Benabdellkader received his MSc degree in computer engineering emphasizing on the integration of heterogeneous-distributed databases, from the Institute of Applied Science of Lyon, France. He joined the University of Amsterdam in 1997; currently, he is going to complete his PhD at the Informatics Institute, Faculty of Science, in the area of integration/interoperation of autonomous data sources in heterogeneous distributed applications. His current research focuses on the design and prototypical development of the information management architecture, the modeling constructs, and the mechanisms to support the tasks of supervision and distributed control in a multi-agent environment.

C. Garita is a senior PhD student at the Computer Science Department of the Faculty of Science of the University of Amsterdam in the Netherlands. In the last years, he has been directly involved in several international R&D projects, focusing on the analysis, design and implementation of federated information management systems to support collaborative scenarios among distributed heterogeneous nodes. In particular, his current research concerns the application of federated database architectures to support the information management requirements set by virtual enterprises in different sectors, such as industrial manufacturing and tourism.

L.O. Hertzberger received the Master’s degree in experimental physics in 1969 and the PhD in 1975, both from the University of Amsterdam. From 1969 to 1983, he was a Staff Member in the High Energy Physics group, later the NIKHEF-H (Dutch Institute for Nuclear and High Energy Physics). In 1983, he was appointed as Professor in Computer Science. His current research interests are in the field of parallel computing, intelligent autonomous robotics and their application in industrial automation.