1 Grid-based Interactive Decision Support in BioMedicine

A. TIRADO-RAMOS[1], P.M.A. SLOOT[1], and M. BUBAK[2,1]

[1] Informatics Institute, University of Amsterdam, Amsterdam, The Netherlands
[2] Institute of Computer Science, AGH, Kraków, Poland

A huge gap exists between what we know is possible with today's machines and what we have so far been able to finish.

—Donald Knuth

1.1 INTRODUCTION

The challenges discovered when studying humans as complex systems, from a biomedical viewpoint (from cells to interacting individuals), cover the whole spectrum from genome to health and cross temporal and spatial scales [1]. This includes studying biomedical issues using multiscale and multiscience models and techniques all the way from genomics to the macroscopic medical scale. This is also aggravated by the continuous increase in the amount of digital data produced by modern high-throughput biomedical detection and analysis systems. As reported by Hey et al., it is expected that larger amounts of digital data will be generated by next generations of large scale, collaborative e-Science experiments [2]. New experiments in science and engineering will cover the whole spectrum, from the simulation of complete biological systems, to cutting-edge research in bioinformatics.

At the macroscopic scale, for instance, there are research efforts in biomedical informatics that are gradually pushing the boundaries of the state of the art, moving from monolitic software architectures to building more generic components. Such efforts normally leverage object-oriented and distributed component architectures to encapsulate or wrap legacy data in order to improve application interoperability and scalability [3, 4]. This allows for enhanced data and process flow at the macroscopic level, where models such as DICOM provide support for data acces from work stations to archiving and communications systems and back to hospitals' information systems.

Current distributed computing technologies address communication among tightly-coupled systems very well, though may fail when addressing loosely-coupled resources. Such resources may belong to sites within large distributed virtual organizations that use distributed computing models like the Common Object Resource Broker Architecture (CORBA) [5]. These technologies allow seamless and secure data access *within a single organization*. Large amounts of data can be distributed across domains, with distributed applications forming federations that may be scaled but that assume architecturally invariant systems. Relational data representation and access, modular component and object oriented models have clearly advanced the state of the art. New sets of conditions and requirements for software architectures for biomedical applications are emerging in the field, particularly in systems biology, where:

- a high degree of user interaction is required
- digitalized biomedical resource usage spaces become increasingly distributed
- inaccessibility and lack of interoperability among modeling, simulation and analysis tools should be leveraged for multi-disciplinary biomedical informatics research to be possible

In this chapter we discuss interactive decision support environments, from the perspective of both health informatics and bioinformatics, for a system-level approach to distributed collaborative laboratories for biomedicine. In the first case we present a problem solving environment for decision support through virtual bypass surgery. In the second case we focus on a more complex system centered around a decision support engine for drug ranking in Human Immunodeficiency Virus (HIV) drug-resistance. Our reasons to use this bioinformatics application as main case study in this chapter are twofold: HIV drug resistance is becoming an increasing problem worldwide, with combination therapy with antiretroviral drugs failing to completely suppress the virus in a considerable number of HIV-infected patients. On the other hand, HIV drug resistance is one of the few areas in medicine where genetic information is widely available and has been used for many years. This has resulted in large numbers of data available, not only on complex genetic sequences but on all levels, up to populations. The sheer complexity of the disease, the distribution of the data, the required automatic updates to the knowledge base, and the efficient use and integration of advanced statistical and numerical techniques necessary to assist the physician motivate our research. We discuss here some of the possibilities for individualized e-Science that can be supported by virtual collaborative environments based on Grid technology [6].

This chapter is divided as follows: in section 1.2 we discuss a Problem Solving Environment (PSE) for biomedical applications on the Grid, focusing on interactive simulation scenarios. In section 1.3 we present research on a collaborative virtual laboratory for e-Science which fulfills the requirements presented by decision support-centric applications. We finalize in section, 1.4 where we present a discussion and future work in the field.

1.2 A GRID FOR INTERACTIVE APPLICATIONS

Processing, visualization and integration of information from various sources play an increasingly important role in modern healthcare [7]. Information sources may be widely distributed, and the data processing requirements can be highly variable, both in the type of resources required and the processing demands put upon these systems. Grid technology is one of the cornerstones of today's computational science and engineering; it offers a unified means of access to different and distant computational and instrumental resources, unprecedented possibilities and benefits are expected. Connectivity between distant locations, interoperability between different kinds of systems and resources, and high levels of computational performance are some of the most promising features of the Grid. In the case of biomedical applications, issues such as remote access to patient data, medical knowledge bases, advanced visualization technologies and specialized medical instruments are of the most importance [8]. For these applications, Grid technology provides dedicated support such as strong security, distributed storage capacity, and high throughput over long distance networks [9]. Besides these immediate benefits, the computational resources of the Grid provide the required performance for large scale simulations, complex visualization and collaborative environments, which are expected to become of major importance to many areas of medicine in order to study the possibilities and limitations of interactive and collaborative problem solving environments.

We propose an architecture for interactive biomedical applications running on a Grid [10], focusing on a simulation-centric application and on production-type Grid infrastructure requirements. We map this architecture to an interactive Problem Solving Environment (PSE) [11, 12] for computer simulation of pre-operative planning of vascular reconstruction being developed by the University of Amsterdam [13, 14], and define a model for representing it that reflects the loosely-coupled and concurrent nature of Grid computing .

Our Grid architecture allows us to build an interactive PSE which offers an integrative approach for constructing and running complex interactive systems on the Grid: highly distributed computational, storage and Grid service resources are used for access to medical image repositories for the simulation and visualization of blood flow. We deployed this interactive PSE within the European CrossGrid framework [15] (Figure 1.1), exploiting available achievements from other European Grid projects such as European DataGrid (EDG) [16] and the Large hedron collider Computing Grid (LCG) [17]. For additional background, motivation, and the latest Grid-based results, we refer the reader to [18].

1.2.1 Interactive Simulation on the Grid

Scalability and seamless resource sharing are at the heart of the Grid-based architectural design, which we base on requirements by a simulation-centric interactive biomedical application, and a production Grid infrastructure and services. We use the Virtual Radiology Explorer (VRE) environment, developed at the University of Amsterdam, is part of a PSE that puts a user at the center of an experimental cycle

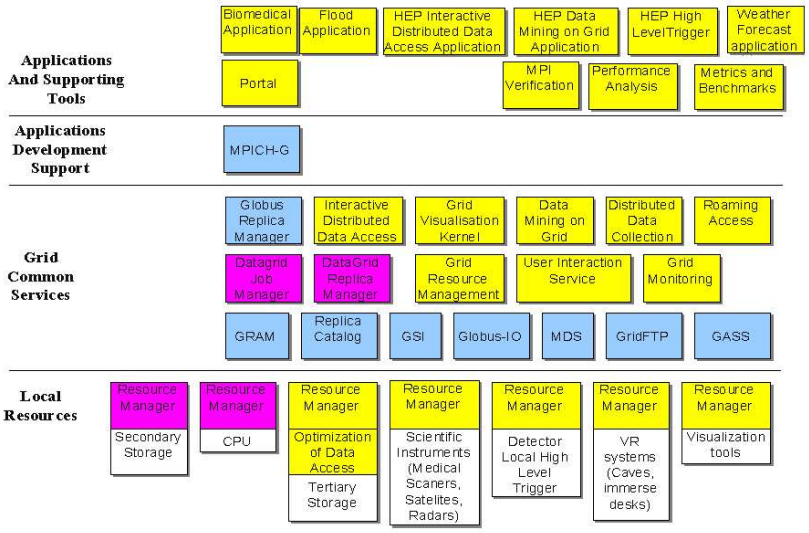

Fig. 1.1 The CrossGrid testbed: layered architectural view, with local resources at the bottom fabric layer, a set of common Grid services and middleware from Globus, DataGrid and CrossGrid, application development support for cross-site job submission, and a set of interactive applications such as the bimedical VRE

controlled by a computer, and allows him to apply his expertise in-silico to find better solutions for treatment of vascular diseases. The aim of the VRE is to provide an end user with an intuitive virtual simulated environment to access medical image data, visualize it, and explore patient vascular condition.

Naturally, since this kind of medical image processing is usually a complicated and resource intensive task, additional computational resources are needed. The VRE contains an efficient parallel computational hemodynamics solver [19] that computes pressure, velocities, and shear stresses during a full systolic period. The simulator is based on the Lattice-Boltzmann method (LBM), a mesoscopic approach for simulating fluid flow based on the kinetic Boltzmann equation [20]. To convert the medical scans into LBM meshes, the raw medical data is first segmented such that only the arterial structures of interest remain in the data. The segmented data is then converted into a mesh that can be used by the LBM solver; boundary nodes, inlet and outlet nodes are added to the Grid using a variety of image processing techniques. The simulator generates the patient blood flow parameters using Grid resources. In order to allow for parallel execution, the simulation volume is divided into several sub-volumes, and each sub-volume is processed concurrently [21]. For visualization the VRE uses a semi-immersive 2-dimensional wall as a projection environment [22, 23, 24].

The VRE system uses a virtual reality environment where the patient's data obtained from the imaging modality is visualized as a 3D stereoscopic image, together

with the graphical interpretation of the simulation results [25]. A user can then manipulate the 3D images of arteries, patient's body and blood flow structures in virtual reality, an environment where users interact freely in a 3D space with entities within it. The working prototype of the VRE is provided with a multi-modal interface described in [26].

1.2.2 Usage Scenario

Once we designed and specified the Grid architecture, we consider the following scenario (Figure 1.2): a patient walks into a Medical Center scanning room somewhere in Europe to get his blood flow measured, the technician scans the abdominal aorta area, and the resulting image is stored in the radiology information system repository or Picture Archiving and Communications System (PACS), to be pre-examined and segmented. Later, a physician (user) somewhere else (e.g., in Amsterdam) logs into the CrossGrid Migrating Desktop (MD) Grid portal using his Grid certificate and private key. The user checks if there are segmented or non-segmented medical data sets ready for analysis and simulation in one of the virtual nodes to work with them locally, and securely transfers a few. The user then starts the DesktopVRE from within the portal, loads the segmented medical data, selects a region of interest, crops image, adds a bypass, and creates a LBM mesh. The user selects the biomedical application icon within the portal (parameters and files are taken from user's profile), and submits the job to the CrossGrid, to the nearest/most adequate CE in the Grid, using a replica manager service. The user may then check job submission or simulation progress via the portal. After the job has been completed, the velocities, pressure, and shear stress are transferred to the local SE or to the appropriate visualization engine to be rendered and reviewed by the user. This scenario implies downloading the portal, from Poznan Supercomputing and Networking Center, to a local roaming storage element; secure access to the testbed; virtual exploration of available SEs throughout the Grid; secure data transfer from an image repository SE (Leiden Medical Center); the preparation of the data for the blood flow simulation within the DesktopVRE version of the biomedical application; job submission to the LBM solver via the RB, at Lisbon Instrumentation and Experimental Particle Physics Laboratory; and visualization of the simulation results using the visualization tools from Johannes Kepler University Linz. All processes are transparent to the user. For more details on integration of the visualization service into the testbed, we refer the reader to [27].

1.2.2.1 Medical Image Segmentation Once medical images are acquired, e.g., by magnetic resonance angiography (MRA), the data is stored in a medical image repository for further analysis. Next, advanced image segmentation techniques are applied: the accurate assessment of the presence and extent of vascular disease requires the determination of vessel dimensions. For this, a method for automatically determining the trajectory of the vessel of interest, the luminal boundaries, and subsequent the vessel dimensions has been developed by the Department of Radiology,

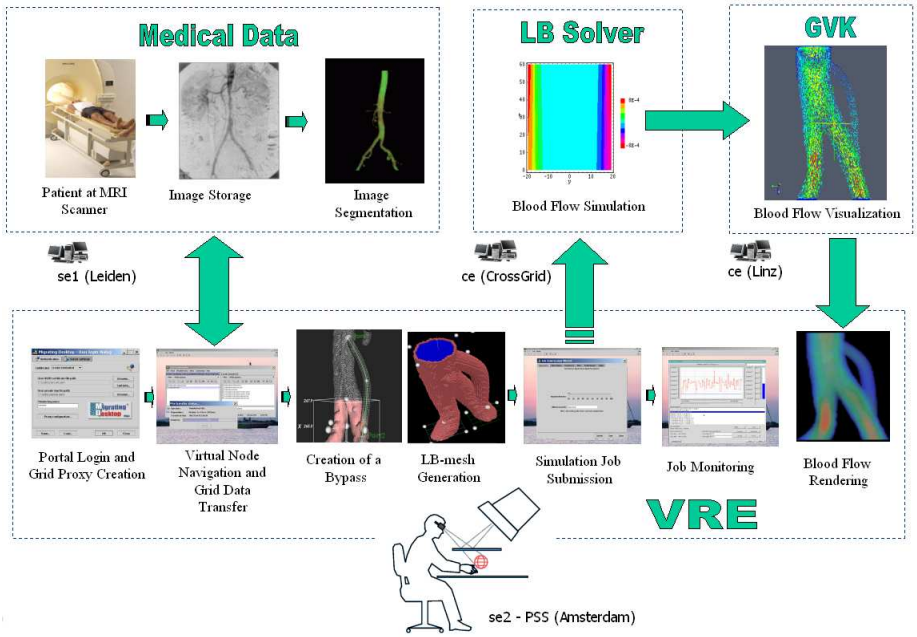

Fig. 1.2 Data and process flow in the VRE environment, where a Grid-based virtual simulated environment is used to access medical image data, visualize it, and explore patient vascular condition; here the Grid-based process flow allows for natural mapping to the user process flow

Leiden University Medical Center (LUMC) [28]. This way, relevant 3D structures such as arteries, are extracted from the raw data.

The Grid portal we use enables the user to access the Grid resources from roaming machines like stand-alone PCs, notebooks or desktop workstations. It allows running applications, managing data files, and storing personal settings independently of the localization or the terminal type. Users may handle Grid and local resources, run applications, manage data files, and store personal settings. The portal provides a front-end framework for embedding some of the application mechanisms and interfaces, and facilitates the user virtual access to Grid resources from other computational nodes. Access to the CrossGrid testbed is based on globus Grid security infrastructure (GSI). GSI uses public key encryption, X.509 certificates, and the secure sockets layer (SSL) communications protocol. Extensions to these standards have been added for single sign-on and delegation. The GSI provides a delegation capability, with an extension of the standard SSL protocol to reduce the number of times the user must enter his pass phrase. If a Grid computation requires that several Grid resources are used (each requiring mutual authentication), or if there is a need to have agents (local or remote) requesting services on behalf of a user, the need to re-enter the user's pass phrase can be avoided by creating a proxy.

Average transfer times for the medical image data, once taking into account the Globus caching mechanism, did not vary much above 400ms for the smaller size files and no more than 850ms for the larger size files.

The segmentation step is connected to the DesktopVRE-based reconstruction of a 3D model of an artery. A geometrical modeling tool allows the interactive manipulation with 3D geometry and procedures, such as the clipping operation, editing of LBM mesh, handling of problematic areas and interactive placement of a bypass (Figure 1.3).

1.2.2.2 Simulation Job Submission Within the portal, application-specific information can be described using extensible markup language (XML) schema. In order to integrate visualization libraries into the computational Grid testbed, we created and posted application XML schemata for job submission, to be dynamically linked to the portal via a job submission wizard. Then XML style sheet transformations (XSLT) is used in order to transform the schemas into appropriate XHTML. The portal sends the job request to the RAS, which then is sent to a job submission service, which then sends the job to a RB and logs all operations. The RB starts a job on the target CE. Before the job is started, a job submission script downloads all necessary files for simulation from a virtual node. Within the portal, we use the EDG replica manager for replication services, which allows one to copy files into Grid storage, register files, replicate files between SEs, delete individual replicas, and delete all replicas of a particular file.

Grid monitoring in CrossGrid includes services for application monitoring, as well as services for monitoring instruments and infrastructure. Application monitoring is substantially different from monitoring infrastructure and instruments, so separate approaches are offered, with application monitoring aimed at observing a particular execution of an application. The collected data is useful for tools for application

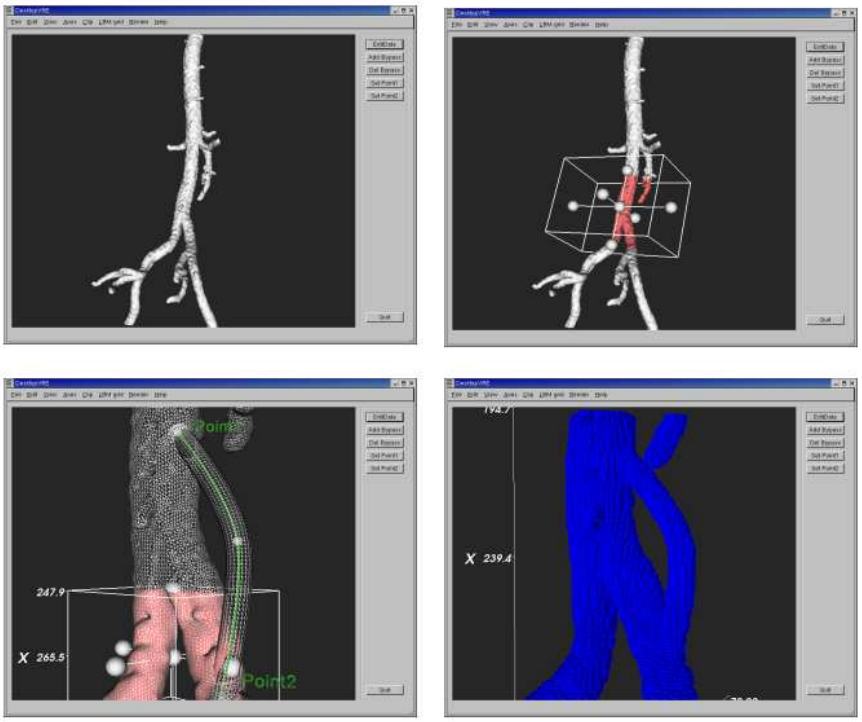

Fig. 1.3 Segmented data rendering, bypass creation, and Lattice-Boltzmann mesh creation in the 2D Desktop Virtual Radiology Explorer; the LBM mesh editing also allows indicating boundary conditions

Fig. 1.4 Simulation monitoring via the CrossGrid light-weight portal, based on an Enterprise Information Portal, using Java and XML, and interfacing with a light-weight Personal Digital Assistant (PDA)

development support, which are used to detect bugs, bottlenecks or just visualize the application's behavior, in the context of a particular execution. For our purposes, we use the MD portal and CrossGrid light-weight portal capabilities for monitoring job submission (Figure 1.4).

We integrated more advanced application monitoring and performance prediction tools, as well as fine-grain infrastructure monitoring to allow for more interactive usage of collected information. The application monitoring infrastructure developed in the CrossGrid is the Grid-based OMIS-compliant Monitoring (OCM-G) [29], a distributed descentralized, autonomous system that runs as a permanent Grid service. It provides monitoring services accesible via a standardized interface, to be used by visual tools such as Grid-Performance Monitoring (GPM)[30]. We use GPM extensively (Figure 1.5) to define our own performance metrics and also to access a set

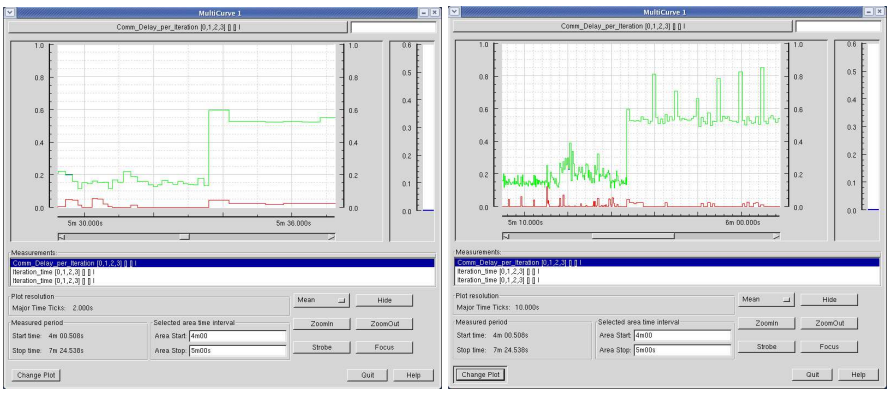

Fig. 1.5 Simulation monitoring of the blood flow simulation in the CrossGrid, via OCM-G/G-PM: graphs showing communication delay per iteration or percentage of time which the solver spends in communication routines in one iteration, at 2 different time-scales of simulation within 5 and 7 minutes of running time (x-axis) and delay (y-axis). The figure identifies a two-fold jump in amount of time in communication at 5.4 minutes on the shorter time scale, which continues on the larger time scale and points out to possible bottlenecks on kernel performance

of fixed ones, and to handle the process of measuring the performance properties. We also use GridBench [31], a tool developed to administer benchmarking experiments, publish their results, and produce graphical representations of their results.

After jobs have been completed, the user can then submit the results for visualization on the Grid using Grid visualization tools, either to anywhere in the network or to specific CE where the simulation has run, to avoid large data transfers. Finally, visualization results are transferred to the VRE to be rendered and reviewed by the user.

We address the combination of Grid applications and corresponding visualization clients on the Grid. While Grids offer a means to process large amounts of data across distant resources, visualization aids in understanding the meaning of data. For this reason, visualization capabilities use Globus services, thereby providing Grid visualization services via dedicated interfaces and protocols while at the same time exploiting the performance of the Grid for visualization purposes. A resource intensive module of the visualization pipeline is instantiated on a high-performance computer. Then, the visualization pipeline on a graphics workstation connects (via re-direction through the portal service) to this module, uses the power of the high-performance computer to generate the visual results, and downloads them to the visualization device. We created links within the MD portal for initialization of the visualization client application, and experimented with rendering the flow both remotely and locally in the access storage element. This way, remote visualization and local rendering are fully linked via the portal, for final rendering.

1.3 INDIVIDUALIZED BIOMEDICAL E-SCIENCE

Computer science provides the language needed to study and understand complex biomedical systems. Computer system architectures reflect the same laws and organizing principles used to build individualized biomedical systems, which can account for variations in physiology, treatment, and drug response.

On the other hand, closing the computational gap in systems biology requires constructing, integrating, and managing a plethora of models. A bottom-up, data-driven approach does not work in this case. Integrating often incompatible applications and tools for data acquisition, registration, storage, provenance, organization, analysis, and presentation requires using Web and Grid services. Once the computational and integration challenges are adressed, we need a system-level approach to close the collaboration and interaction gap [32]. Such an approach involves sharing processes, data, information, and knowledge across geographic and organizational boundaries within the context of distributed, multidisciplinary, and multi-organizational collaborative teams, or virtual organizations. These methods dynamically streamline and, most importantly, *individualize* scientific data flow processes depending on their availability, reliability, and the specific interests of medical doctors, surgeons, clinical experts, researchers, and other end users We call this a *molecule to man* approach (Figure 1.6).

1.3.1 The ViroLab Collaboratory

During the past decade, researchers have made significant progress in treating patients with viral diseases. Effective antiretroviral therapy has lead to sustained HIV viral suppression and immunological recovery in patients who have been infected with the virus. Adherence to antiretroviral treatment, therefore, remains the cornerstone of effective treatment, and failure to adhere is the strongest predictor of virological failure. Long-term therapy can lead to metabolic complications. Other treatment options are now available, with the recent introduction to clinical practice of fusion inhibitors, second-generation nonnucleoside reverse transcriptase inhibitors, and nucleotide reverse transcriptase inhibitors. However, in order to completely suppress the virus, patients must take a combination of at least two of the four different classes of antiretroviral drugs [33]. Nevertheless, in a significant proportion of patients the drugs fail to completely suppress the viral disease, resulting in the rapid selection of drug-resistant viruses and loss of drug effectiveness. This complicates the clinician's decision process, since clinical interpretation is based on data sets relating mutations to changes in drug sensitivity and relating mutations present in the virus to clinical responses to specific treatment regimens.

In recent years, researchers have developed several genotypic resistance-interpretation tools that help clinicians and virologists choose effective therapeutic alternatives to address, e.g., genotypic resistance interpretation. Furthermore, applying artificial intelligence and computational techniques has resulted in the development of specialized computer-based Decision Support Systems (DSSs). Recent developments in distributed computing further allow the virtualization of the deluge of available data,

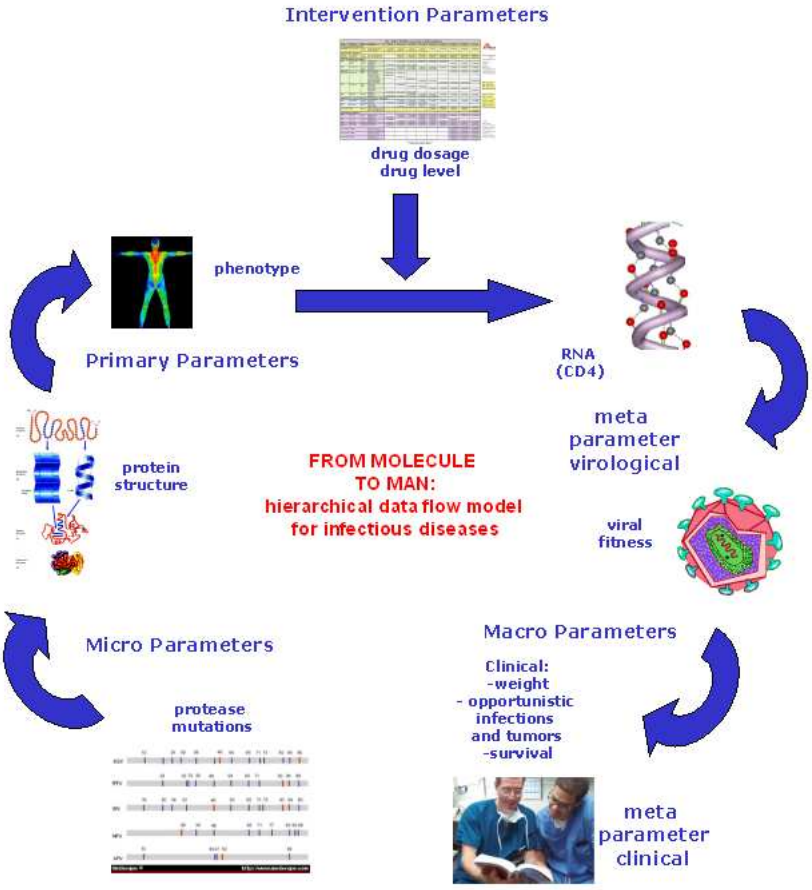

Fig. 1.6 Data flow model in ViroLab, showing the model for prediction of the temporal behaviour of the immune system to drug therapy aims to qualitatively correspond to clinical data. The multiscale approach from micro parameters such as protease mutations to macro results at the clinical level go through primary, interventional and meta virological parameters, as supported by the Virtual Laboratory

computational and software resources required by complex e-Science. ViroLab[34] is an international collaborative laboratory which goal is to provide such virtual laboratory where researchers and medical doctors have easy access to distributed simulations and can share, process, and analyze virological, immunological, clinical, and experimental infectious disease data [35]. Currently, virologists browse journals, select results, compile them for discussion, and derive rules for ranking and making decisions.

ViroLab's Grid-based DSS for infectious diseases consists of modules for individualized drug ranking in human immunodeficiency disease. It offers clinicians a distributed virtual laboratory securely accessible from their hospitals and institutes throughout Europe.

1.3.2 System Requirements

ViroLab's research goal is to investigate novel computational methods and techniques that support the development of a secure and user-friendly integrated decision support system for physicians. We use emerging Grid technology to combine data discovery, data mining, statistical analyses, numerical simulation and results presentation.

Some of the main technical requirements for building such a system include:

- efficient data management
- integration and analysis
- error detection
- recovery from failures
- monitoring and logging for process flows
- distributed execution of data- and compute-intensive tasks
- visualization and image processing on the data through the analysis steps
- metadata-based data access, authentication, and authorization

For the system to support Grid-based distributed data access and computation, virtualization of its components is important. ViroLab includes advanced tools for biostatistical analysis, visualization, modeling, and simulation that enable prediction of the temporal virological and immunological response of viruses with complex mutation patterns for drug therapy [36, 37].

1.3.2.1 Decision Support System A DSS and data analysis tools are at the center of the ViroLab distributed collaboratory. Such tools may estimate the sensitivity for available drugs by interpreting a patient's genotype using mutational algorithms that experts developed based on scientific literature, taking into account the published data relating genotype to phenotype. This way, rankings are also based on data from clinical studies of the relationship between the presence of particular mutations and the clinical or virological outcome.

A number of bioinformatics software programs have been developed in the last few years to support bioinformatics decision making in clinical environments. A

couple of examples of such systems are the Virtual Phenotype (developed by Virco NV)[1], and Retrogram (developed by Virology Networks BV[2]). The output of these programs consists of a prediction of the drug sensitivity of the virus, generated by comparing the viral genotype to a relational database containing a large number of phenotype-genotype pairs. The Retrogram[3] decision software, in particular, interprets the genotype of a patient by using rules developed by experts on the basis of the literature, taking into account the relationship of the genotype and phenotype [38]. In addition, it is based on (limited) available data from clinical studies and on the relationship between the presence of genotype directly to clinical outcome. It is important to note, however, that these systems focus on biological relationships and are limited to the role of resistance. The next step is to use clinical databases and investigate the relationships between the viral resistance profile (mutational profile and/or phenotypic data) and therapy outcome measures such as amount of virus (HIV-RNA) and CD4+cells.

In DSSs like the Retrogram, the primary goal of the data analysis is to identify patterns of mutations (or naturally occurring polymorphisms) associated with resistance to antiviral drugs and to predict the degree of in-vitro or in-vivo sensitivity to available drugs from an HIV genetic sequence. The statistical challenges in doing such analyses arise from the high dimensionality of these data. A variety of approaches have been developed to handle this type of data, including clustering, recursive partitioning, and neural informatics. Neural informatics is used for synthesis of heuristic models received by methods of knowledge engineering, and results of the formal multivariate statistical analysis in uniform systems. Clustering methods have been used to group sequences that are near each other according to some measure of genetic distance: once clusters have been identified, recursive partitioning can be used to determine the important predictors of drug resistance, as measured by in-vitro assays or by patient response to antiviral drugs.

1.3.2.2 Interactivity and Process Flow The availability of Grid infrastructures and tools for interactive applications presents an important research opportunity. For bioinformatics collaboratory work we build on previous efforts where we developed a unified approach for running interactive distributed applications on the Grid by providing solutions to the following issues:

- automatic porting of applications to Grid environments
- user interaction services for interactive startup of applications, online output control, parameter study, and runtime steering
- advanced user interfaces that enable easy plug-in of applications and tools, like interactive performance analysis combined with online monitoring
- scheduling of distributed interactive applications

[1] http://www.vircolab.com/
[2] http://www.vironet.com/
[3] Registered Trademark University of Amsterdam: no. 713908

- benchmarking and performance prediction
- optimization of data access to different storage systems

In ViroLab, an important issue is for users to be able to register and publish derived data and processes and to keep track of the provenance of information flowing through the generated pipelines, as well as accessing existing (patient and scientific literature) data and acquiring new data from scientific instruments. These domain-independent features can then be customized by adding domain-specific components and semantic annotation of the components and data being used. In order to automate the construction of process flow applications, the system needs to generate ontological descriptions of services, system components, and their infrastructure [39, 40]. Semantic data is usually stored as a registry that contains Web Ontology Language (OWL) descriptions of service class functionality, instance properties, and performance records. The user provides a set of initial requirements about the process flow use, then the system builds an abstract process flow using the knowledge about services' functionality that service providers have supplied to the registry. Subsequently, the system must apply semantic information on service properties, which results from analyzing the monitoring data of services and resources, to steer running process flows that still have multiple possibilities of concrete Web service operations.

The system can select the preferable service class by comparing semantic descriptions of the available services classes and matching the classes' features to the actual requirements. ViroLab users can therefore verify and identify the data's origin and rerun experiments when required. ViroLab extends this feature by categorizing the level of information, including the data and process flows. The collected data-provenance information is archived in ViroLab's portal and accessible through search and discovery methods.

1.3.2.3 Virtual Organization Grid computing is based on the central concept of distributed *Virtual Organizations* that span multiple trust domains. Trust in Grids is commonly established via a Public Key Infrastructure (PKI): every entity in the system is issued with a "certificate" that links an identifier to a piece of unique cryptographic data.

In Virolab we developed a distributed virtual organisation that binds the various components of the distributed VO. This binding layer spans a number of geographically separated physical institutions across Europe, including five hospitals. ViroLab's VO-based security infrastructure is based on Grid middleware and a set of interfaces providing user-friendly and transparent access to the ViroLab applications, within a Grid portal.

Security is, naturally, an important concern. The sensitive nature of clinical patient data, together with concerns that data and resources be made available in a timely fashion to just those who are authorized to access them, is supportd by Grid authentication and authorization components which span all aspects of the infrastructure. It is important to note, though, that in ViroLab Grid security policy definition is left to the local owner's trust policy. VO members with access to the

VOs resources can therefore use and share distributed resources securely, leveraging single-sign on.

Maintaining confidentiality is important in the development of monitoring protocols and procedures. Therefore, in order to guarantee patient confidentiality, database access is limited and anonymized, especially in the case of overview in which HIV results can be linked to individuals. In this case such overviews are destroyed as soon as relevant data have been retrieved.

1.3.3 System Architecture

The ViroLab system's design guarantees the interaction between a user and running applications, similar to methods used in real experiments, so the user can change a selected set of input data or parameters at runtime. For instance, under a typical usage scenario in ViroLab:

- a scientist from a clinical and epidemiological virology laboratory in Utrecht, Netherlands securely accesses virus sequence, amino acid, or mutations data from a hospital AIDS lab in Rome using Grid technology components running in Stuttgart, Germany.
- the scientist applies quality indicators needed for data-provenance tracking using provenance-server components running in Krakow, Poland.
- researchers use this data is used as input to (molecular dynamics) simulations and immune system simulations running on Grid nodes that reside at University College London and the University of Amsterdam.
- the virtualized DSS automatically derives meta rules.
- intelligent system components from Amsterdam use first-order logic to clean rules, identify conflicts and redundancy, and check logical consistency.
- the scientist validates new rules that the system automatically uploads into the virtualized DSS.
- the system presents a new ranking.

We next elaborate on ViroLab's Grid-based architecture design, in terms of the system's virtual laboratory, presentation and virtualization viewpoints (Figure 1.7).

Our system's architecture is based on the Grid concept of distributed virtual organizations (VOs), which has a virtualized decision support system at its core, and a pervasive Grid infrastructure that provides PKI security access. We distinguish a base *Grid Resources* layer, where computational (computing elements within hospitals or research centers) and data resources (storage elements where individual patient data, medical knowledge data, intermediate experimental data, and so forth) is archived. On top of this layer, a virtual laboratory component encapsulates the runtime system that interacts with the collaboration and data access components via a session manager that handles as weel the provenance components. Finally, an application and presentation layers contain the user interfaces and individual application interfaces to either the core rule-based system used for initial decision support and ranking, or the scientific

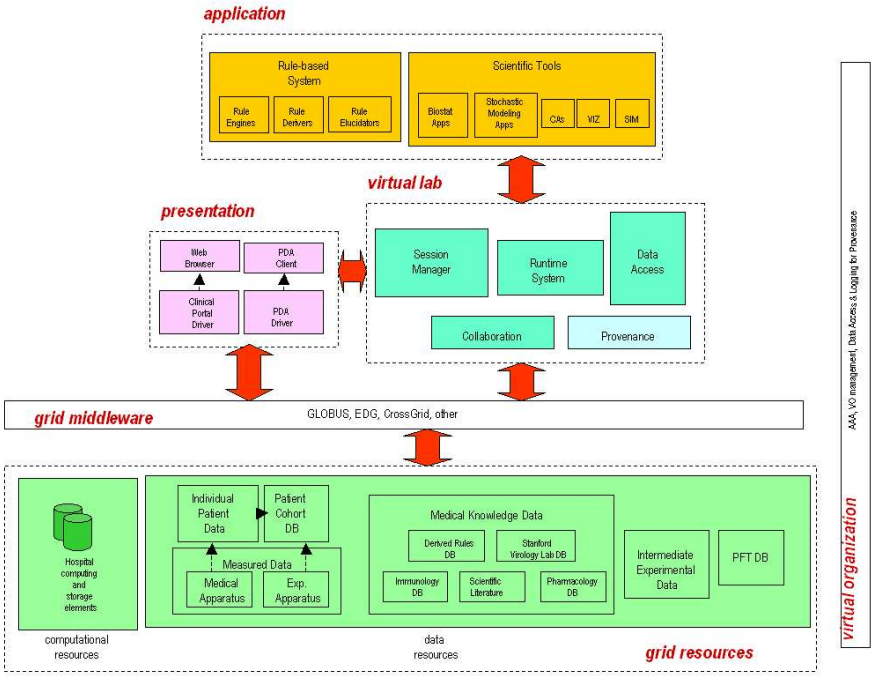

Fig. 1.7 ViroLab system architecture; distributed resources (computing elements, data, and storage) that the biomedical applications use are coordinated with the Grid middleware and a virtualized runtime system. Resources are automated and virtualized, and the resulting data is fed to anonymizing components, as well as directly to the Decision Support System

tools for the enhancement of such rankings. We next elaborate on the different components and their functionalities.

1.3.3.1 Virtual Laboratory In order to cover the temporal and spatial scales required to infer information from a molecular level up to patient medical data, multi-scale methods are applied in ViroLab, where distributed simulation, statistical analysis and data mining are combined and used to enhance the base rule-based decision process. In this scenario, resources are widely distributed, and the data processing requirements are highly variable, both in the type of resources required and computational processing demands. Experiment design, integration of information from various sources, as well as transparent scheduling and execution of experiments is initially supported by the DAS2 testbed in The Netherlands, providing additional computational power for computational intensive jobs. We reuse Grid middleware from successful European projects to provide basic Grid services for data management, resource management, and information services on top of Globus middleware.

In order to support such a distributed decision support infrastructure, we start with VO support for a DSS-centered prototype. Here, users are assisted by rules developed by experts on the basis of available literature, taking into account the relationship between relevant genotype and phenotype data. We extend this monolithic base DSS by virtualizing its basic building blocks, and distributing the relevant components and data from clinical studies across the VO.

1.3.3.2 Presentation In the initial design phases we aim to maintain the same level of usability and readability as the original Web version of the DSS interface. This is accomplished by maintaining the same structure, but with some modifications. A Proxy method is implemented for accessing the web-based software from mobile devices as well, where the Proxy server acts between the remote server (the DSS) and a mobile device. Here, a navigation script in the Proxy is responsible for the following:

- take the patient data from the mobile user (i.e. patient detail, laboratory information)
- create an HTTP communication with the remote server
- submit data to the remote server
- take the result from the remote server
- parse HTML code and retrieve only relevant information (i.e. drug ranking, error messages, drug references etc.)
- send the wireless pages to the mobile device

In the initial Web version, the Proxy is implemented using a hypertext preprocessor as a server-site scripting language running on the Web server (Figure 1.8). Two versions are developed using the Proxy method: WAP version and web clipping. If a user wants to enter the patient details fields, he has to move from one screen to the other and come back again. The fields already filled in the previous screens should not be lost.

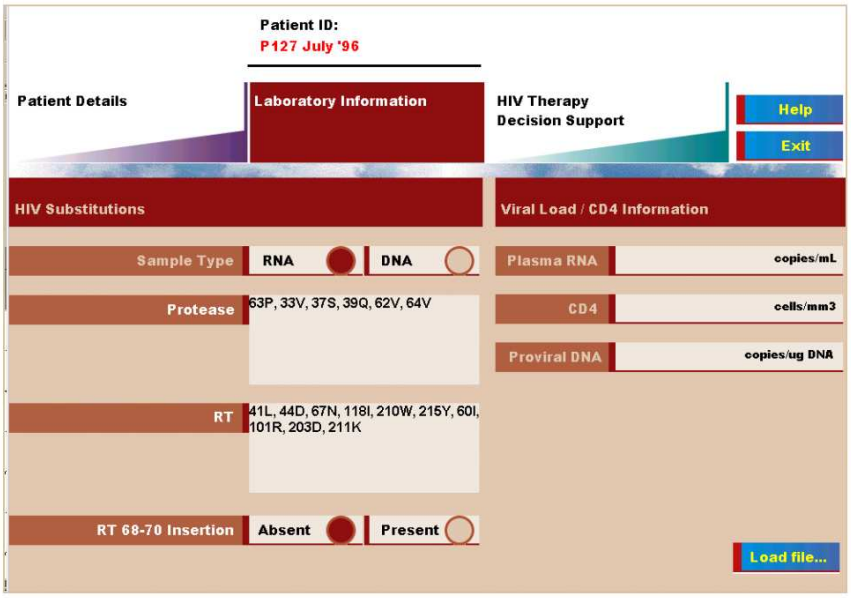

Fig. 1.8 Web Retrogram: User enters patient substitutions in order to get drug ranking results by login into the interface, accessing patient detail and laboratory information, and accessing the DSS for drug ranking and interpretation, courtesy of P.M.A. Sloot, A.V. Boukhanovsky, W. Keulen, A. Tirado-Ramos and C.A. Boucher, "A Grid-based HIV Expert System", *Journal of Clinical Monitoring and Computing*, October 2005, vol. 19, nr. 4-5, pp. 263-78

For ViroLab, an extra layer of Grid services is implemented in order to allow access to both applications and resources via a Grid portal. The portal serves as the central access point where users are authenticated using single sign-on, and provides direct access to the virtual laboratory infrastructure, runtime system and collaboration support. Our aim is that the portal is based on standard portlet technologies, using a set of portlet web applications that collaborate within the framework and support standard Grid security. We initially leverage the support for Grid integration of the GridSphere portal framework. In GridSphere, a collection of Grid portlets provided as add-on modules form a cohesive end-user environment for managing users, groups and supporting remote job execution, file staging and providing access to information services. GridSphere provides two portlet implementations; one is the JSR 168 de facto portlet API standard and the other is based upon the IBM WebSphere Portlet API. GridSphere supports the development of re-usable portlets and portlet services. It includes a set of core portlets and portlet services that provide the basic infrastructure required for developing and administering Web portals.

1.3.3.3 Application and Resource Virtualization ViroLab infrastructure provides virologists with an advanced environment to study trends on an individual, population, and epidemiological level. That is, by virtualizing the hardware, compute infrastructure, and databases, the virtual laboratory offers a user-friendly environment, with tailored process flow templates to harness and automate such diverse tasks as data archiving, integration, mining, and analysis; modeling and simulation; and integrating biomedical information from viruses (proteins and mutations), patients (viral load), and the literature (drug-resistance experiments).

In ViroLab we need access to different types of data resources via Grids. In order to achieve this goal, we provide a way of querying, updating, transforming and delivering data via web services in a consistent, independent way. In order to automate archiving, integration, mining, as well as transparent access to applications, we work with metadata and the data resources in which this data is stored, accessed via web services that can be combined to provide higher-level services that support Grid-based data federation and distributed query processing.

We approach virtualization by allowing data and application resources, to be accessed via web services. That is, a web service allows data to be queried, updated, transformed and delivered, while integrating the data via services to clients. In ViroLab we use the OGSA-DAI web services model, which can be deployed within a Grid environment. This allows us to Grid-enable the distributed data resources.

1.4 DISCUSSION AND SUMMARY

With the increasing availability of genetic information and extensive patient records, researchers can now study diseases from the DNA level all the way up to medical responses. Resolving the long-standing challenges of individual-based, targeted treatments is coming within reach. It is necessary to provide integrating technology

to the medical doctors and researchers bridging the gaps in multiscale models, data fusion, and cross-disciplinary collaboration.

With the first case study we showed a framework for rapid prototyping of exploration environments that permits users to explore interactively the visualized results of a simulation, and manipulate the simulation parameters in near real-time. We introduced a generic architectural requirements, defined a generic component-based software architecture and its abtract interactions, and identified a specific use case for validation of the architecture.

The second and main case study presented here consists of a decision support system compares patients' viral genotype to a distributed relational database containing a large number of phenotype-genotype pairs. The decision software interprets a patient's genotype by using rules developed by experts on the basis of the literature, taking into account the relationship of the genotype and phenotype. In addition, the output is based on available data from clinical studies and on the relationship between the presence of genotype and the clinical outcome.

We have showed how in the understanding of processes from bioionformatics to heath informatics, from molecule to man, distributed computing in general and Grid technology in particular can play a crucial role. We found that in order to cover the huge time and spatial scales required to infer information from a molecular (genomic) level up to patient medical data, we need to apply multi-scale methods where simulation, statistical analysis, data-mining is combined in an efficient way. Moreover, such required integrative approach requires distributed data collection (e.g. HIV mutation databases, patient data, literature reports, etc.) and a virtual organization (physicians, hospital administration, computational resources, etc.) to support it. The access to and use of large-scale computation (both high performance as well as distributed) is essential since many of the computations involved require near real-time response and are to complex to run on a personal computer or personal digital assistant. Furthermore, data presentation is crucial in order to lower the barrier of actual usage by the physicians, here the Grid technology (server-client approach) can play an important role.

For future work, we will work on enhancements to the current testbed like scientific collaboration support, as required to process the variety of data and information generated from a number of ViroLab applications as well as data providers and hospitals. For instance, in addition to the basic requirements of voice and video support between scientists, we will work on scientific collaboration support for the sharing of drug rankings (current rankings and new rankings resulting from the new applications), collaborative validation of drug rankings (once validation of a new ranking has been performed, users may want to discuss and share their findings with relevant stakeholders), and feedback from experts via links to the workflow engine (collaboration tools may allow the direct and instant communication with experts during and at all steps of scientific workflow execution).

Acknowledgments

The authors wish to thank the members of the CrossGrid and ViroLab consortia. This research is partly funded by the EU IST CrossGrid (IST200132243) and ViroLab (IST027446), and the Bsik Virtual Laboratory for eScience (VL-e) projects.

REFERENCES

1. Anthony Finkelstein, James Hetherington, Linzhong Li, Ofer Margoninski, Peter Saffrey, Rob Seymour, Anne Warner, Computational Challenges of System Biology, IEEE Computer, May 2004, Vol. 37, No. 5, pp. 26-33.

2. A.J.G. Hey and A.E. Trefethen, The Data Deluge: An e-Science Perspective, F. Berman, G.C. Fox, and A.J.G. Hey, eds., Grid Computing Making the Global Infrastructure a Reality, Wiley and Sons, 2003, pp. 809-824.

3. R. Martinez; J.F. Cook and A. Tirado-Ramos: Java CORBA DICOM Adapter Service for DICOM-compliant Datasets, in First Latin American Conference on Biomedical Engineering, pp. 621-622. Sociedad Mexicana de Ingenieria Biomedica IEEE-EMBS, Mazatlan, Mexico, November 1998. ISBN 968-5063-03-6.

4. A. Tirado-Ramos; J. Hu and K.P. Lee: Information Object Definition-based Unified Modeling Language Representation of DICOM Structured Reporting: A Case Study of Transcoding DICOM to XML, Journal of the American Medical Informatics Association, vol. 9, pp. 63-72. 2002.

5. Object Management Group, The Common Object Request Broker: Architecture and Specification, 2.0 ed., July 1995.

6. Foster, I., C. Kesselman, and S. Tuecke, The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of High Performance Computing Applications, 2001, 15(3), pp. 200-222.

7. S. Martin, D. C. Yen, J. K. Tan, E-health: impacts of internet technologies on various healthcare and services sectors, International Journal of Healthcare Technology and Management 2002, Vol. 4, pp. 71-86.

8. P.M.A. Sloot, Simulation and visualization in medical diagnosis: Perspectives and computational requirements, in A. Marsh, L. Grandinetti, T. Kauranne (Eds.), Advanced Infrastructures for Future Healthcare, IOS Press, 2000, pp. 275-282.

9. M. Cannataro, C. Comito, A. Congiusta, G. Folino, C. Mastroianni, A. Pugliese, G. Spezzano, D. Talia, and P. Veltri. Grid-based PSE Toolkits for Multidisciplinary Applications. FIRB Grid.it WP8 Working Paper 2003-10, ICAR-CNR, December 2003.

10. A. Tirado-Ramos; P.M.A. Sloot; A.G. Hoekstra and M. Bubak, An Integrative Approach to High-Performance Biomedical Problem Solving Environments on the Grid, Parallel Computing, special issue on High-Performance Parallel Biocomputing, Chun-Hsi Huang and Sanguthevar Rajasekaran editors, 2004, vol. 30, nr 9-10 pp. 1037-1055.

11. E.N. Houstis, J.R. Rice, S. Weerwarna, P. Papachio, Wang K. Yang, M. Gaitatzes, Enabling Technologies for Computational Science frameworks middleware and environments, Kluwer Academic Publishers, 2000, Chapter 14, pp. 171-185.

12. Gallopoulos, E., Houstis, E., and Rice, J. R. 1994, Computer as Thinker/Doer: Problem-Solving Environments for Computational Science, IEEE Comput. Sci. Eng., Volume 1, Issue 2, June 1994, pp. 11-23.

13. K.A. Iskra; R.G. Belleman; G.D. van Albada; J. Santoso; P.M.A. Sloot; H.E. Bal; H.J.W. Spoelder and M. Bubak: The Polder Computing Environment, a system for interactive distributed simulation, Concurrency and Computation: Practice and Experience, (Special Issue on Grid Computing Environments), John Wiley and Sons, vol. 14, 2002, pp. 1313-1335.

14. R.G. Belleman, P.M.A. Sloot, Simulated vascular reconstruction in a virtual operating theatre, CARS 2001 Conference, Berlin, Germany, June 2001.

15. M. Bubak, M. Malawski, and K. Zajac, Architecture of the Grid for Interactive Applications, Proc. Int. Conf. Computational Science LNCS 2657, Springer, 2003, pp. 207-213; www.crossGrid.org.

16. EU DataGrid Project, http://eu-datagrid.web.cern.ch/eu-datagrid/

17. http://lcg.web.cern.ch/LCG/Documents/default.htm

18. http://www.science.uva.nl/research/scs/HotResults/

19. A.M. Artoli, A.G. Hoekstra, P.M.A. Sloot, 3D Pulsatile flow with the Lattice Boltzmann BGK method, International Journal of Modern Physics C, Volume 13, Issue 08, 2002, pp. 1119-1134.

20. S. Succi, The Lattice Boltzmann Equation for Fluid Dynamics and beyond, Oxford Science Publications, Clarendon Press, 2001.

21. A.M. Artoli, A.G. Hoekstra, P.M.A. Sloot, Mesoscopic simulations of systolic flow in the human abdominal aorta, Journal of Biomechanics, vol. 39, 2006, pp. 873-884.

22. R.G. Belleman, B. Stolk, R. de Vries, Immersive virtual reality on commodity hardware, in: Proceedings of the Seventh Annual Conference of the Advanced School for Computing and Imaging, May 2001, pp. 297-304.

23. The University of Amsterdam Distributed Real-time Interactive Virtual Environment, UvA-DRIVE http://www.science.uva.nl/ robbel/DRIVE/

24. E.V. Zudilova, P.M.A. Sloot, R.G. Belleman, A Multi-modal interface for an interactive simulated vascular reconstruction system, in: Proceedings of the IEEE International Conference on Multimodal Interfaces, Pittsburgh, Pennsylvania, USA, October 2002, pp. 313-319.

25. R.G. Belleman, J.A. Kaandorp, D. Dijkman, P.M.A. Sloot, GEOPROVE: geometric probes for virtual environments, Lecture Notes in Computer Science, no. 1593 Springer-Verlag, Berlin , April 1999, pp. 817-827.

26. E.V. Zudilova, P.M.A. Sloot, Virtual reality and desktop as a combined interaction-visualisation medium for a problem-solving environment, in: Proceedings of International Conference on Computational Science-ICCS 2003, Melbourne, Australia, in series Lecture Notes in Computer Science, Springer Verlag, June 2003, pp. 1025-1035.

27. A. Tirado-Ramos, H. Ragas, D. Shamonin, H. Rosmanith, D. Kranzlmueller, Integration of blood flow biomedical application Grid visualization Kernel via a Grid portal, Second European Across Grids Conference, Nicosia, Cyprus, 28-30 January 2004.

28. P.F. Lelieveldt, S.C. Mitchell, J.G. Bosch, R.J. vander Geest, M. Sonka, J.H.C. Reiber, Time continuous segmentation of cardiac image sequences using active appearance motion models, proc. IPMI, Lect. Notes Comput. Sci. 2082 (2001), pp. 446-452.

29. B. Balis, M. Bubak, W. Funika, T. Szepieniec, R. Wismueler, An infrastructure for Grid application monitoring, in: D. Kranzlmueller, P. Kacsuk, J. Dongarra, J. Volker, (Eds.), Recent advances in parallel virtual machine and message passing interface, Proceedings of the Ninth European PVM/MPI Users Group Meeting, Linz, Austria, September/October 2002, LNCS 2474, 2002, pp. 41-49.

30. Marian Bubak, Wlodzimierz Funika, Roland Wismueller, Tomasz Arodz, Marcin Kurdziel, The G-PM Tool for Grid-Oriented Performance Analysis, Lecture Notes in Computer Science, Volume 2970, Jan 2004, pp. 240-248.

31. Tsouloupas, G. Dikaiakos, M.D., GridBench: a tool for benchmarking Grids, Proceedings. Fourth International Workshop on Grid Computing, pp. 60-67, 2003.

32. P.M.A. Sloot; A. Tirado-Ramos; I. Altintas; M.T. Bubak and C.A. Boucher; From Molecule to Man: Decision Support in Individualized E-Health, IEEE Computer, November 2006, vol. 39, no. 11, pp. 40-46.

33. S.G. Deeks, Treatment of Antiretroviral-Drug-Resistant HIV-1 Infection, Lancet, 13 Dec., 2003, pp. 2002-2011.

34. http://www.virolab.org/

35. P.M.A. Sloot; A.V. Boukhanovsky; W. Keulen; A. Tirado-Ramos and C.A. Boucher: A Grid-based HIV Expert System, Journal of Clinical Monitoring and Computing, October 2005, vol. 19, nr. 4-5, pp. 263-78.

36. P.M.A. Sloot, F. Chen, and C.A. Boucher, Cellular Automata Model of Drug Therapy for HIV Infection, S. Bandini, B. Chopard, and M. Tomassini, eds., Proc. 5th Intl. Conf. Cellular Automata for Research and Industry (ACRI 02), Springer, LNCS 2493, Springer, 2002, pp. 282-293.

37. T.E. Scheetz et al., Gene Transcript Clustering: A Comparison of Parallel Approaches, Future Generation Computer Systems, May 2005, pp. 731-735.

38. Been-Tiktak A, Korn K, Keulen W, Schwingel E, Walter H, Schmidt B, et al. Evaluation of an open expert-based genotype interpretation program: RetroGram. 41st Interscience Conference on Antimicrobial Agents and Chemotherapy, 2001.

39. Bertram Ludascher, Ilkay Altintas, Chad Berkley, Dan Higgins, Efrat Jaeger, Matthew Jones, Edward A. Lee, Jing Tao, Yang Zhao, Scientific workflow management and the Kepler system, Concurrency and Computation: Practice and Experience, 2006, vol. 18, no. 10, pp. 1039-1065.

40. Marian Bubak, Tomasz Gubala, Michal Kapalka, Maciej Malawski and Katarzyna Rycerz, Workflow Composer and Service Registry for Grid Applications, Future Generation Computer Systems, Volume 21, Issue 1, January 2005, Pages 79-86.