1

A Virtual Laboratory for Medical Image Analysis

Sílvia D. Olabarriaga, *Member, IEEE,* Tristan Glatard, Piter T. de Boer

***Abstract*—This paper presents the design, implementation and usage of a virtual laboratory for medical image analysis. It is fully based on the Dutch Grid, which is part of the EGEE production infrastructure and driven by the gLite middleware. The adopted service-oriented architecture enables decoupling the user-friendly clients running on the user's workstation from the complexity of the grid applications and infrastructure. Data is stored on grid resources and can be browsed/viewed interactively by the user with the Virtual Resource Browser (VBrowser). Data analysis pipelines are described as Scufl workflows and enacted on the grid infrastructure transparently using the MOTEUR workflow management system. VBrowser plug-ins allow for easy experiment monitoring and error detection. Thanks to the strict compliance to grid authentication model, all operations are performed on behalf of the user, ensuring basic security and facilitating collaboration across organisations. The system has been operational and in daily use for 8 months (Dec 2008), with 6 users, leading to the submission of 9,000 jobs/month in average and the production of several terabytes of data.**

***Index Terms*—Medical image analysis, grid, workflow management, gLite, e-Science, SOA, grid usability.**

I. Introduction

MEDICAL imaging research is experiencing the "data deluge" phenomenon also observed in other sciences. The resolution and number of images are increasing, and the scale of studies are growing to involve multiple centers, requiring infrastructure beyond conventional solutions provided by a researcher's workstation. Additional capacity is required to store and analyze the data, and new means are needed to support and facilitate collaboration to exchange data, analysis methodology, and knowledge. Properly organizing an IT infrastructure to perform large imaging studies tends to require significant effort that distracts the researchers (and funds) from the original imaging research question. Take the example of functional Magnetic Resonance Imaging (fMRI [1]), which is a popular tool used in neuroscience research to study brain activation. Large amounts of data (images and signals) are acquired, processed, compared, annotated, and potentially shared by many users that typically collaborate across institutional boundaries (e.g., a university and a hospital). Although some groups have been capable of organizing a shared infrastructure to facilitate data analysis and archival (e.g. BIRN[1]), in many cases researchers still depend heavily on their personal computers and skills to collect and analyze data from various sources. Grids, on the other hand, can potentially address some of these problems, by facilitating access to a shared distributed system that provides high-performance computing, storage and services to users organized in "virtual organizations" (VOs) [2].

The grid for medical imaging described here has its origins in the Virtual Laboratory for e-Science (VL-e[2]) project, where concepts of grids and e-Science are exercised in various application domains. One of the first pilots in the medical subprogram of VL-e (vlemed) aimed at addressing difficulties for analysis and management of fMRI data (see details in [3]). It was coined "virtual lab for functional MRI" because the grid infrastructure is presented as a natural extension of the "laboratory" available to the fMRI scientist, including the MRI scanner, computers, and analysis software. After 3 years of progressive development and improvement, the system initially conceived for fMRI evolved into a more generic infrastructure, now coined "virtual lab for medical imaging", which is daily adopted autonomously by end-users mainly for neuroimaging research.

In this paper we present an overview of this virtual lab, describing the system evolution, design and architecture in section II. Two application examples are presented in section III to illustrate the usage and performance of the virtual lab for research in functional MRI and Diffusion Tensor Imaging (DTI). Usage statistics demonstrate the relevance of our approach. Related work is discussed in section IV, and a discussion, conclusions and future prospects close the paper.

II. System Description

The virtual lab provides hardware, software and services to support data storage, analysis, and collaboration in large scale medical imaging studies. The available tools facilitate data collection at the scanning site (scanner, EEG, etc.); storage of large amounts of data; high-throughput computing via parallel analysis of mutually independent data; remote access to all resources via a user-friendly interactive interface; and sharing of data and services. The system was initially designed to address the needs of fMRI data analysis and management, taking advantage of the grid infrastructure available for the VL-e project (details in [3]). From the start, the goal was to put the infrastructure at service of end-users with various

Manuscript created December 9, 2008, revised May 20, 2009, and accepted December 6, 2009. This work was funded by the Virtual Laboratory for e-Science project (www.vl-e.nl), which is supported by a BSIK grant from the Dutch Ministry of Education, Culture and Science (OC&W) and is part of the ICT innovation program of the Ministry of Economic Affairs (EZ).

S.D.Olabarriaga is with the Department of Clinical Epidemiology, Biostatistics and Bioinformatics, Academic Medical Center of the University of Amsterdam, Postbus 22660, 1100 DD Amsterdam, NL (corresponding author: +31 20 566 4660, S.D.Olabarriaga@amc.nl). T. Glatard was with the Department of Clinical Epidemiology, Biostatistics and Bioinformatics, Academic Medical Center of the University of Amsterdam, NL. He is now with the University of Lyon, CNRS, INSERM, CREATIS, 69621 Villeurbanne, FR (glatard@creatis.insa-lyon.fr). P.T. de Boer is with the Informatics Institute, University of Amsterdam, Science Park 909, 1098 XH, Amsterdam, NL and with PCC UvA bv, NL (ptdeboer@uva.nl).

[1]http://www.nbirn.net
[2]http://www.vl-e.nl

backgrounds who regularly perform image acquisition at the Academic Medical Center of the University of Amsterdam (AMC).

A. System evolution

Since 2005, the system evolved through roughly three phases described in detail in [4].

In the first phase ("low hanging fruit"), the main idea was to reuse software components developed in the EGEE project. gLite Linux command line utilities were used from gLite User Interface (UI) systems to perform basic operations such as data transfer between the hospital and the external storage resources, and submission of image analysis jobs. This set-up was never accepted by the end-users because it was too invasive and complex to use. In particular the separation between the user's desktop and "the grid" were inappropriate in imaging research, where data need to be available for visual inspection at all times.

In the second phase ("trying out") several implementation alternatives were investigated using different front-ends and software/middleware stacks. The goal was to improve the usability without sacrificing generality. The interface with data resources was significantly improved with the introduction of the Virtual Resource Browser (VBrowser[3] [5]) which is an interactive tool that enables browsing (grid-enabled) resources from a single application (see section II-E). Pilot fMRI analysis experiments were performed in various ways, first using `edg-*` commands, then Python clients from regular desktops, a workflow management system (VLAM-G [6]) and a parameter sweep engine (Nimrod-G [7], see details in [8]). In all cases, the set-up was still considered too difficult by the end-users because it involved diverse systems and required considerable effort and skills to perform experiments. Moreover, the required connectivity with grid servers was not always possible due to the hospital's security policy.

In the third and current phase ("user ready"), the virtual lab was improved w.r.t. the front-end for job management, connectivity with grid resources, and system reliability. A SOA-oriented approach was adopted to decouple the functionality and complexity (service) from the front-end (client), resulting in lightweight software and reduced connectivity requirements for the user's workstation. The VBrowser was extended to provide access to services, enabling end-users to directly manipulate data and perform analysis using grid resources seamlessly. Additionally, the image analysis experiments were implemented as workflows that are automatically, transparently and reliably enacted on the grid by the MOTEUR engine [9]. Finally, a grid gateway was installed at the hospital to facilitate connectivity with the public grid. This system is described in the following sections.

B. Hardware Infrastructure

The hardware components are illustrated in Fig. 1. The acquisition devices (e.g., MRI scanner) are located at the AMC Radiology network, and the computing and storage resources

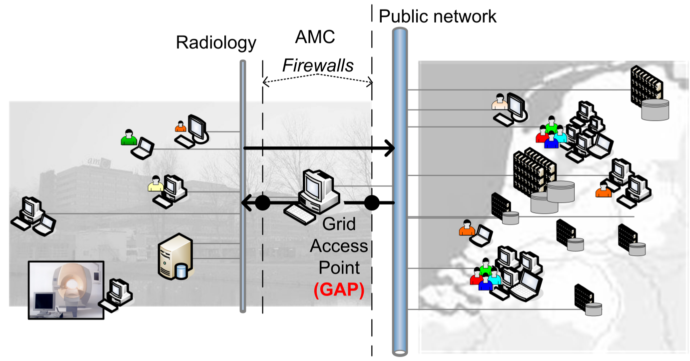

Fig. 1. Hardware components: acquisition devices at the AMC, distributed grid resources, and network gateway. Users access resources from anywhere.

are provided by the Dutch e-Science infrastructure[4]. Users located anywhere (e.g., at the hospital or at home) can access the system to manipulate data and run workflows from regular computers with sufficient outbound connectivity[5].

The Grid Access Point (GAP) facilitates connectivity between the hospital and the public grid, allowing inbound connections from selected hosts to storage resources and services located inside the hospital. Although inbound connectivity is not needed for normal grid usage with the VBrowser, it is necessary to deploy systems that directly link the hospital resources with the grid, such as the Medical Data Manager (MDM [10]). In this manner, DICOM images could be shared as a gLite grid storage and accessed from grid jobs as illustrated in [11]. Note however that the MDM is not used in production yet.

C. Granting access to the system

The virtual lab adopts the VO mechanism to grant shares of system usage, including access to computing, storage and services. Users need to be in possession of a valid certificate[6] emitted by a trusted Certification Authority (CA) and registered in a VO. A Registration Authority has been established inside the hospital to intermediate the communication with the CA, facilitating end-users requests. A special VO (vlemed) was created for the medical image analysis community of the VL-e project, but the system can be configured to authorize any VO.

This authentication/authorization process is still considered complex by end-users. In particular getting started (obtaining a certificate and properly installing it) seems to present much difficulty. Once this is done, most users experience usage as simple. An alternative would be to let the users authenticate on the grid with the same credentials used in their own organisations using Shibboleth[7], e.g., as in [12]. Trust across

[3] http://www.vl-e.nl/vbrowser

[4] See http://www.biggrid.nl. Resources are distributed among 11 sites (Dec 2008), including large computing service providers and small clusters in other organizations, e.g., academic hospitals.

[5] In addition to http(s) ports 80 and 443, also 30003 (vlemed VOMS server), 8443 (Grid services) and 2811/2000x (GridFTP) need to be open for outbound connections. No inbound connections are required

[6] See http://www.globus.org/security/overview.html

[7] http://shibboleth.internet2.edu/

domains remains a barrier for the actual implementation of such mechanisms, especially in hospitals.

D. System Architecture

The virtual lab adopts a layered service-oriented architecture illustrated in Fig. 2. The user interacts with the VBrowser from his/her own workstation (section II-E). General high-level functions, such as workflow management and job monitoring, are implemented as Web and Grid services deployed outside the hospital and managed by the vlemed VO. These services interface with low level grid services and middleware, hiding from the user the heterogeneity and complexity of the infrastructure.

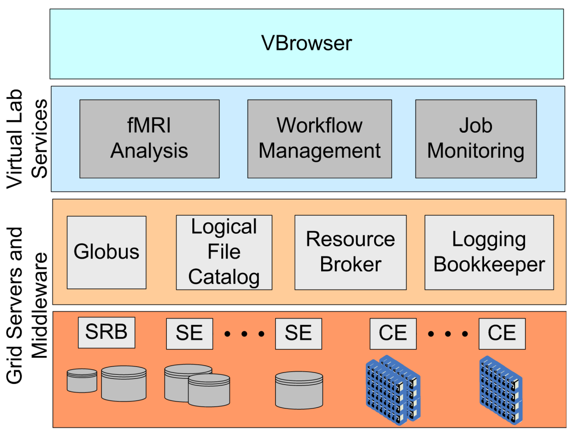

Fig. 2. Virtual lab components: front-end, services and middleware.

The set-up is flexible to enable access to grid resources via different middleware. The application (or user) may access resources directly, e.g., using Globus[8] job submission and data interfaces. In general, however, services rely on gLite because it provides an abstraction layer that simplifies access to the heterogeneous and dynamic pool of resources. In gLite, sites expose Computing Elements (CEs) with queues for job submission federated by a Resource Broker. Similarly, sites expose Storage Elements (SEs) for basic file management, e.g., via GridFTP, and also a Storage Resource Manager (SRM) as a data scheduling layer. A Logical File Catalog (LFC) associates logical file names (LFNs) to physical files stored on the SEs, also supporting file replication and aliasing. When CEs and SEs are added to the infrastructure, or temporarily become off-line, they are automatically and transparently managed.

E. VBrowser

The front-end with the system is provided by the VBrowser, which enables users to access local and remote resources (data and services) from a user-friendly GUI (Fig. 4). More than an interactive tool, the VBrowser is also an integrating platform for application porting that offers an abstract layer for interfacing with a variety of middlewares (Fig. 3). Fully developed in Java, it has its own (optimized) Virtual File System (VFS) that integrates *Virtual Drivers* in a Java API to transparently access files on several storage systems. Files can be copied with one single method from/to any location, independently from the data transfer protocol involved. A file path consists of a unique resource identifier (URIs[9]) containing complete information, such as host and file name. File operations are performed on behalf of the user, ensuring proper authentication and authorization, with grid credentials and/or SSL passwords being managed automatically. Current VFS drivers support local files, sFTP, GridFTP, Globus Reliable File Transfer Service (GT 4.0 RFTS), SRM, San Diego Storage Resource Broker (SRB [13]) and LFC.

The VBrowser can be extended by plug-ins to implement clients for specific services, drivers for various transfer protocols, or viewers for specific data types. Plug-ins can be associated with MIME-types that can be mapped to file extensions, and automatically started to open a document. Currently there are viewers for text (with editing capabilities), HTML, PDF, GIF, and medical images (NIfTI format[10]). Additionally, any locally available application can be triggered as an external viewer, the data being automatically downloaded by the VBrowser. The complete environment (software, credentials, and preferences) can optionally be installed on a memory stick, enabling the user to connect to the grid with the same look-and-feel from any Microsoft Windows, Linux or Macintosh system.

F. Distributed data management

The data facilities of the virtual lab are predominantly used to support large experiments, providing storage federation with an uniform view of local and distributed storage resources, and fault-tolerance to cope with the dynamic nature of the infrastructure. Uniform access to heterogeneous storage resources is obtained by adopting the VFS to implement data access in all applications and services. A specific VFS driver developed for the LFC wraps gLite command-line utilities and includes extra fault-tolerance features such as robust SE selection and automatic file replication. Although users only see LFNs such as `/grid/vo/sharedData/scan.nii.gz`, replicas can be distributed in several SEs. The execution of jobs depending

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

[9] URIs follow the RFC 2396 syntax: http://www.ietf.org/rfc/rfc2396.txt

[10] http://nifti.nimh.nih.gov

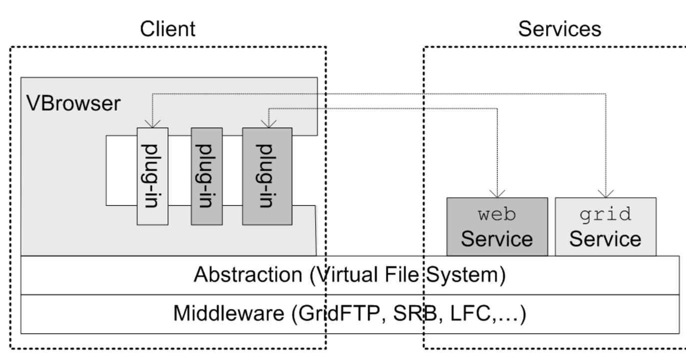

Fig. 3. Software architecture: middleware, abstract layer, VBrowser GUI, remote services, and respective clients as VBrowser plug-ins.

on the file will not be hampered if an SE is temporarily offline and at the same time avoiding bottlenecks on the data server. If the gLite utilities are not installed in the front-end, the driver automatically executes in “service mode”, connecting to a remote GT4 service. This maps LFNs to Transport URLs (TURLs), for example GridFTP, enabling data transport between an SE and regular workstations, e.g. Microsoft Windows, without the need of installing additional (gLite) software besides the VBrowser.

Security is an important aspect for handling medical data, acknowledged as a challenge for the deployment of health-grid [14], which has been only partially addressed in the virtual lab. We currently assume a research scenario where the user manually exports the files to grid storage, e.g., with drag-and-drop on the VBrowser, and configures the access rights using standard GSI and VO authentication and authorization mechanisms. In the LFC, however, access control is performed at a coarse level, only distinguishing user and VO to set file permissions, and easy to bypass when physical file locations are known. Finer granularity could be be implemented using the Virtual Organization Membership Service (VOMS) [15] to define groups and roles, however the dynamicity of data sharing and the high demands on privacy may require more sophisticated authorization mechanisms such as proposed in [16]. The SRB supports user-level access control lists (ACL), currently providing an interesting alternative for secure data sharing. Specifically for medical images in DICOM, we are currently evaluating a prototype MDM service at the AMC [11], since it provides user-level ACL to image header fields and data, on-the-fly de-identification, and encryption/decryption. However, due to remaining security constraints, in particular trust by the AMC network administrators, the MDM is not fully exploited yet for data sharing in our infrastructure. Other alternatives would be Globus-based solutions of the Medicus [17], the German MEDIGRID [18] and TRENDCADIS [19] projects. Note, however, these solutions are specific for DICOM data, which is not adopted in the majority of the compute-intensive studies that have been performed in the virtual lab so far.

G. *fMRI service*

A specific service (*Feat Runner*) was developed to run parameter sweep experiments for fMRI analysis with the fMRI Expert Analysis Tool (`feat`) of the fMRIB Software Library (FSL [20]). The client, implemented as a VBrowser plug-in, provides a GUI where the user specifies input data, parameter values, and ranges for these inputs. The service takes care of launching jobs for each combination of parameter and data values, organizing all related information (inputs, outputs, errors, job ids, etc.) into directories. An HTML summary enables the user to browse all files using mnemonic identifiers associated with the parameter values used in the sweep. This service currently only handles a simple fMRI analysis pipeline for a clinical application in the AMC (neurosurgery planning). A generic application porting/execution layer is implemented with workflow technology as described below.

H. *Computation via workflows*

Image analysis is traditionally implemented by pipelines, so workflows represent a natural approach to run such applications on grids. Additionally, workflows facilitate parallelization of the application and improves reusability and sharing of applications and components that otherwise would be implemented by scripts [21].

In the virtual lab the Scufl language of the Taverna workbench [22] is adopted to describe workflows because (1) it separates the functional description of the workflow (built by the developer) from the data (instantiated by the user); (2) it has a user-friendly graphical representation and (3) iterations over a large number of inputs can be easily described in the language, which is useful to express medical imaging computations. Other task-graph workflow approaches typically used in grid computing (e.g., DAGMan[11], Pegasus [23] or GWES [24]), are more suitable for performance, but they do not separate functional description from data. Other workflow languages such as MoML (used by Kepler [25]) might constitute interesting alternatives to the Scufl language, however they typically lack powerful iteration strategies.

The MOTEUR workflow management engine [9] is adopted to run workflows because it supports gLite for job submission. Additionally MOTEUR has the following interesting features: automatic packaging of legacy applications in workflow components for execution on grids with the Generic Application Service Wrapper (GASW) [26]; redirection of data by reference among components; generation of unique file names for workflow outputs; data provenance logging; fault-tolerance mechanisms (resubmission and timeout); and job grouping to improve usage of the grid resources.

For integration in the virtual lab, we developed a MOTEUR Web-Service and the corresponding VBrowser plug-in. The life cycle starts with the creation of a workflow document (Scufl format) with the Taverna Workbench, to be stored anywhere on a VFS-accessible location. By opening the document in the VBrowser (Fig. 4-1), the MOTEUR plug-in is activated and the user can specify the inputs of the experiment (Fig. 4-2). Ranges or lists are supported to facilitate sweeping on parameter values, and VFS files can be directly dragged-and-dropped into input boxes. When the user presses “run”, the plug-in contacts the MOTEUR service over an HTTPs connection, sending the Scufl description, the input data to be processed (values and URIs), and the user credentials. The MOTEUR service instantiates an engine that submits and monitors jobs with gLite middleware on behalf of the user. The workflow status is maintained through an HTML summary (Fig. 4-3), enabling the user to monitor the execution within the VBrowser. To allow users to monitor the jobs individually and retrieve output sandboxes, we developed a job monitoring grid service and plug-in (Fig. 4-4). Standard output/error of jobs can be downloaded and checked by the user (Fig. 4-5). Since all I/O operations are implemented with the VFS, all files involved in a workflow execution (workflow description, inputs, monitoring and results) are directly available for inspection with the VBrowser (Fig. 4-6).

[11] http://www.cs.wisc.edu/condor/dagman

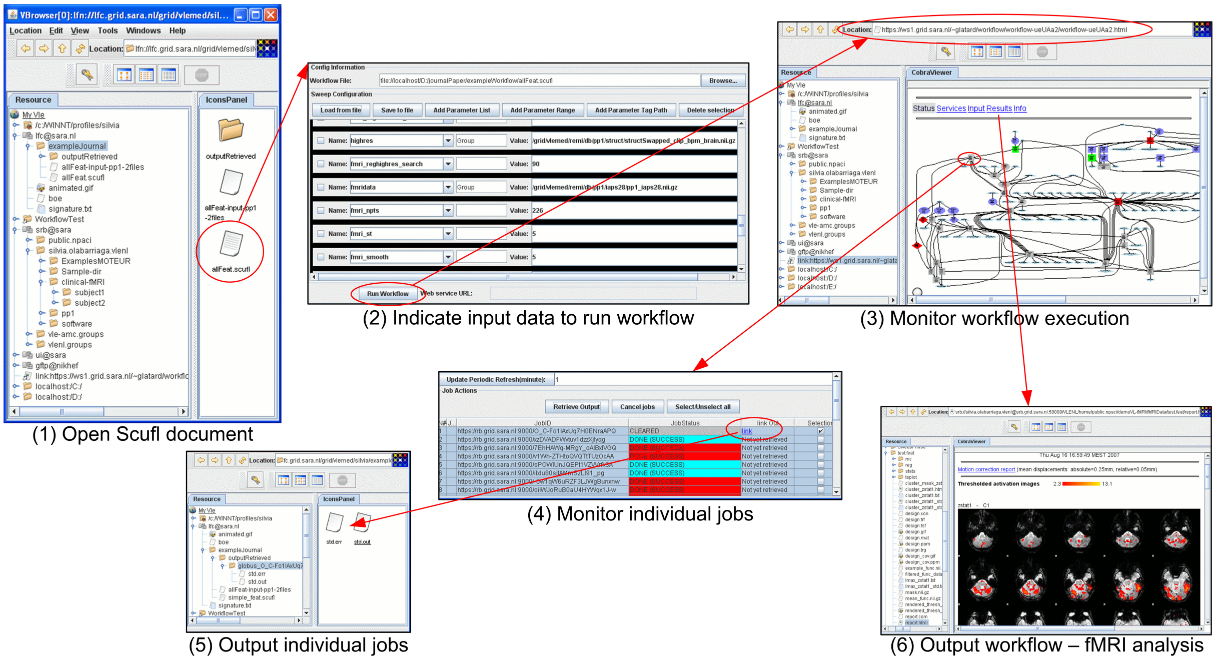

Fig. 4. Steps for the execution and monitoring of workflows on the Dutch grid using MOTEUR and the VBrowser.

III. Results

The virtual lab has been in regular use since March 2008 by a community of 6 users. They have been adopting the system autonomously, from workstations at the hospital, university or home. The largest usage so far is reported by a neuroscientist at the AMC investigating how selected parameters for the fMRI analysis methods influence the computed brain activation map [27]. This experiment required a computation effort of 1 year of CPU and generated 1.5 TB of data in one week. This user now autonomously performs additional experiments to investigate other brain regions and parameters. A medical imaging researcher at the AMC also adopted the virtual lab to validate new DTI image analysis methods developed in Matlab[12]. The matlab code is compiled and executed on the grid using the Matlab Compiler Runtime (MCR) library. From an initial set-up using a campus grid with the help of a computer scientist [28], the experiment is now run in the virtual lab directly by the end-user. Experiments that would take a week to compute sequentially can be completed in a few hours. Recently the virtual lab has also been adopted by bioinformaticians at the AMC for analysis of high-throughput DNA sequencing data [29], as well as by computer scientists for research on scientific workflows for neuroimaging [30] and for optimization tasks [31].

Table I presents usage statistics since the system became operational (8 months). A total of 726 workflows were run, corresponding to the execution of 68,564 tasks (workflow components) on the grid. Note that the number of submitted jobs $(103,748)$ is larger due to the fault tolerant mechanism of MOTEUR, which retries failed jobs up to a given number of times in addition to the resubmissions automatically performed by gLite. Although the success rate for jobs is 61.7%, from the user's perspective the success rate is 86.7%, indicating that the system can self-recover and hide errors due to transient infrastructure problems. The remaining errors are caused by faulty infrastructure components (reported the grid administrators) or by errors in the workflow components or inputs (reported to users and developers). Currently permanent errors are handled manually by the users and the vlemed administrator, and require significant effort and expertise.

[12]http://www.mathworks.com

TABLE I
Activity statistics: number of workflows; workflow tasks (total number, failed and success ratio); submitted grid jobs; and success ratio.

		Workflow tasks			Jobs	
Month	# wf	total	failed	ok (%)	submitted	ok (%)
Mar	147	19645	6544	66.7	36432	36.0
Apr	64	20275	585	97.1	28111	70.0
May	13	3271	256	92.17	3638	82.9
Jun	78	1578	55	96.51	3667	41.53
Jul	89	2471	509	79.4	3941	49.8
Aug	135	7953	1996	74.9	12760	46.7
Sep	114	6969	656	90.0	8493	74.0
Oct	86	6402	222	96.0	6706	92.0
Total	726	68564.0	10823.0	-	103748.0	-
Mean	90.8	8570.5	1352.9	86.7	12968.5	61.7

IV. Related work

The adoption of grids for medical imaging has been an active topic of research. Early projects include, for example, e-Diamond [32], the Loni Pipeline [33], MIAKT [34], IXI [35], NeuroGrid [36], GEMSS [37], Mammogrid [38], Neurobase [39] and DentGrid [40]. More recent projects such as caBIG's GridCAD [41], NeuroLOG [42], Globus

Medicus [43], MEDIGRID [44], CVIMO[13] In particular the funcLAB/G [45] implements fMRI data analysis in a clinical setting using a SOA based on Globus and Medicus. More examples are found in the proceedings of the DIDAMIC'2004 [46] and MICCAI-Grid'2008 [47] workshops. It is difficult, however, to objectively compare these projects to the virtual lab because they vary largely in scope, approach and resources. Considering the infrastructure, most of the above projects (exceptions MammoGrid, NeuroLOG and MEDIGRID) are based on (semi) dedicated resources, whereas the virtual lab adopts an open grid linked to EGEE, which is the largest production infrastructure available today. Respective to the grid middleware, most of them are directly based on Globus (exceptions MammoGrid, NeuGrid, NeuroLOG), whereas the virtual lab mainly adopts a true open grid middleware, gLite, which provides high level brokering for data and computing resources. The virtual lab is similar to many projects that adopt SOA, e.g., GEMSS, funcLAB/G, Mammogrid. Respectively to the front-end, most projects implement custom portals [48] or dedicated analysis applications [49] that hide the grid complexity from the users, trading generality for usability. For example, in the funcLAB/G a fixed fMRI analysis pipeline is automatically executed in an easy manner, and in the Health e-Child project [50] a user-friendly front-end implements a dedicated clinical application. Although this is a convenient approach for the end-users, it requires a large development effort for each new application. In the virtual lab the front-end is generic, supporting data analysis and management from a GUI without constraining the user to predefined usage scenarios. Examples of other generic user-friendly front-ends for grid-enabled applications are g-Eclipse [51], the Migrating Desktop [52], and the P-Grade portal [48]. Perhaps the distinguishing factor between the VBrowser and these frameworks is the tight connection between (virtual) resources and the corresponding viewers. This facilitates the integration of user interfaces for grid applications and services in a natural manner on the user's workstation, as if these were extensions of their local computational resources. Users are still free to choose how to organize and manipulate data, modify and run workflows (e.g. with FSL or compiled matlab), and inspect results in their preferred manner.

V. Conclusions

The virtual lab presented in this paper successfully integrates many existing e-Science tools and platforms into a high-level user-friendly system. To achieve this goal, several services and VBrowser plug-ins have been developed by our project to enable LFC access, fMRI experiments, workflow executions with MOTEUR and job monitoring. Integration has been performed with a strong concern for preserving reusability (via workflows), security (strict conformance to the grid security model), scalability (based on grid middleware) and usability (VBrowser GUI). As a result, the developed virtual lab for medical image analysis offers a very rapid start-up for running experiments on the EGEE production grid.

The data and computing resources are distributed in The Netherlands and constantly being upgraded, but the adoption of gLite and the VBrowser interface hides it all. Thanks to the workflow approach, new grid applications are made available by the developers as simple documents that are open with the same interface. Apart from the proxy generation that has to be done by the user through the VBrowser, the whole grid configuration is delegated to the MOTEUR service. An administrator can then configure the workflow execution parameters without any burden for the user. Nevertheless, getting a certificate and joining the VO is still tiresome and could be facilitated with better user interface and support.

The system has been operational since March 2008, being currently autonomously used by 6 end-users for research. Since that time, about 9,000 jobs per month have been run in average. To our knowledge, such statistics of autonomous usage of grids by end-users in medical image analysis are the first published. Autonomy has greatly increased the usage of the system: users are no longer afraid of trying out new analyses or running large experiments, since these no longer require any external commitment from grid experts. As a side effect, we observe that most of the users are not aware of the scale and complexity of the system they use. In particular, we notice that the expectations concerning response time do not match the reality of a highly distributed system with so many dispersed components. Since users simply "double-click" on documents to access the virtual lab functions, they expect similar response time as for resources on their desktop. Faster networks are likely to improve response time, but the weight of intermediate software layers needed to "simply open a file" will remain.

Future work includes the exploitation of collaborative scenarios for sharing data, algorithms and workflows, and the development of automatic strategies for error detection and management. The virtual lab is currently being tested and deployed by other communities, e.g. at the AMC for DNA sequencing and in other European projects in medical image analysis[15], which will certainly foster new exciting challenges in (meta-)data, computing and experiment management.

Acknowledgment

We warmly thank our closest end-users: A. Nederveen, R. Soleman, M. Caan and A. Luyf. We are also grateful in particular to Kamel Boulebiar for his invaluable contribution to the VBrowser plug-ins development. We also thank other VL-e colleagues who collaborate(d) in the design, development and operation of the virtual lab (J.J. Keijser, D. van Dok, M. Bouwhuis, B. Heupers, K. Maheshwari); the AMC network administrators; and essential support teams of the BiG Grid project, in particular the anonymous administrators behind *grid.support@sara.nl* and *grid.support@nikhef.nl*.

References

[1] S. Smith, "Overview of fMRI analysis," *Br J Radiology*, vol. 77, pp. S167–S175, 2004.

[13]http://www.grycap.upv.es/cvimo, and NeuGrid[14] are trying to progress beyond the level of demonstrators and deploy grid systems for medical imaging in research and clinical environments.

[15]http://neurolog.polytech.unice.fr, http://www.creatis.insa-lyon.fr

[2] I. Foster, C. Kesselman, and S. Tuecke, “The anatomy of the grid: Enabling scalable virtual organizations,” *Int J Supercomputer Applications*, vol. 15, p. 2001, 2001.

[3] S. D. Olabarriaga, A. J. Nederveen, J. G. Snel, and R. G. Belleman, “Towards a Virtual Laboratory for fMRI Data Management and Analysis,” in *Proceedings of Healthgrid 2006*. IOS Press, 2006, pp. 43–54.

[4] S. Olabarriaga, T. Glatard, K. Boulebiar, and P. T. de Boer, “From ’low-hanging’ to ’user-ready’: initial steps into a healthgrid,” in *Proceedings of Healthgrid 2008*. IOS Press, 2008, pp. 70–79.

[5] S. Olabarriaga, P. de Boer, A. Belloum, K. Maheshwari, J. Snel, A. Nederveen, and M. Bouwhuis, “Virtual Lab for fMRI: Bridging the Usability Gap,” in *2nd IEEE Int Conf e-Science and Grid Computing*, 2006.

[6] A. Belloum *et al.*, “VLAM-G: a Grid-based Virtual Laboratory,” *Future Gener Comput Syst*, vol. 19, no. 2, pp. 209–217, 2003.

[7] R. Buyya, D. Abramson, and J. Giddy, “Nimrod-G Resource Broker for Service-Oriented Grid Computing,” *IEEE Dist Systems Online*, vol. 2, no. 7, 2001.

[8] S. Olabarriaga, A. J. Nederveen, and B. O. Nuallain, “Parameter Sweeps for Functional MRI Research in the Virtual Laboratory for e-Science Project,” in *7th IEEE Int Symp Cluster Computing and the Grid*, May 2007, pp. 685–690.

[9] T. Glatard, J. Montagnat, D. Lingrand, and X. Pennec, “Flexible and efficient workflow deployement of data-intensive applications on grids with MOTEUR,” *Int J High Performance Computing and Applications*, vol. 22, no. 3, pp. 347–360, Aug. 2008.

[10] J. Montagnat *et al.*, “A Secure Grid Medical Data Manager Interfaced to the gLite Middleware,” *J Grid Computing*, 2007.

[11] “Image of the week - Medical Data Manager,” *Int Science Grid This Week (iSGTW)*, Oct. 2008. [Online]. Available: http://www.isgtw.org/?pid=1001439

[12] R. O. Sinnott, J. Jiang, J. Watt, and O. Ajayi, “Shibboleth-based access to and usage of grid resources,” in *7th IEEE/ACM Int Conf Grid Computing*, 2006, pp. 136–143.

[13] C. Baru, R. Moore, A. Rajasekar, and M. Wan, “The SDSC storage resource broker,” in *CASCON’98 Conference*, 1998.

[14] M. Olive, H. Rahmouni, and T. Solomonides, “SHARE the journey: A European Healthgrid Roadmap,” Tech. Rep., 2008.

[15] R. Alfieri, R. Cecchini, V. Ciaschini, and F. Spataro, “From gridmap-file to VOMS: managing authorization in a Grid environment,” *Future Gener Comput Syst*, vol. 21, pp. 549–558, 2005.

[16] I. Blanquer, V. Hernandez, D. Segrelles, and E. Torres, “Enhancing Privacy and Authorization Control Scalability in the Grid Through Ontologies,” *IEEE Trans Inform Technol Biomed*, vol. 13, no. 1, pp. 16–24, 2009.

[17] S. Erberich, M. Bhandekar, A. Chervenak, and C. Kesselman, “DICOM grid interface service for clinical and research PACS: a globus toolkit web service for medical data grids,” *Int. J. Computer Assisted Radiology and Surgery*, vol. 1, pp. 87–105, June 2006.

[18] M. Vossberg, T. Tolxdorff, and D. Krefting, “DICOM Image Communication in Globus-Based Medical Grids,” *IEEE Trans Inf Technol Biomed*, vol. 12, pp. 145 – 153, 2008.

[19] I. Blanquer Espert, V. Hernández García, and D. Segrelles, “TRENCADIS - A WSRF Grid Middleware for Managing DICOM Structured Reporting Objects,” in *Proceedings of HealthGrid 2006*. IOS Press, 2006, pp. 381–391.

[20] S. Smith *et al.*, “Advances in functional and structural MR image analysis and implementation as FSL,” *NeuroImage*, vol. 23, pp. 208–219, 2004.

[21] K. Fissell, “Workflow-based approaches to neuroimaging analysis,” in *Neuroinformatics*, ser. Methods in Molecular Biology, November 2007, vol. 401, pp. 235–266.

[22] T. Oinn *et al.*, “Taverna: A tool for the composition and enactment of bioinformatics workflows,” *Bioinformatics J*, vol. 17, no. 20, pp. 3045–3054, 2004.

[23] E. Deelman *et al.*, “Mapping Abstract Complex Workflows onto Grid Environments,” *J Grid Computing*, vol. 1, no. 1, pp. 9–23, 2003.

[24] A. Hoheisel, “Grid workflow execution service - dynamic and interactive execution and visualization of distributed workflows,” in *Cracow Grid Workshop 2006*, 2006.

[25] B. Ludäscher *et al.*, “Scientific Workflow Management and the Kepler System,” *Conc and Comput: Practice & Experience*, vol. 18, no. 10, pp. 1039 – 1065, Aug. 2006.

[26] T. Glatard, J. Montagnat, D. Emsellem, and D. Lingrand, “A Service-Oriented Architecture enabling dynamic services grouping for optimizing distributed workflows execution,” *Future Gener Comput Syst*, vol. 24, no. 7, pp. 720–730, 2008.

[27] T. Glatard *et al.*, “Large-scale functional MRI study on a production grid,” *Future Gener Comput Syst*, vol. In Press, pp. –, 2010.

[28] S. Caton *et al.*, “Using dynamic condor-based services for classifying schizophrenia in diffusion tensor images.” in *7th IEEE Int Symp Cluster Computing and the Grid*, 2008, pp. 234–241.

[29] B. D. van Schaik *et al.*, “Handling next generation sequence data: a pilot to run data analysis on the Dutch life science grid,” in *Benelux Bioinformatics Conf.*, 2008.

[30] T. Glatard and S. Olabarriaga, “From gridified scripts to workflows: the FSL Feat case,” in *Workshop on Medical Imaging on Grids*, 2008, pp. 33–42.

[31] A. Wibisono, Z. Zhao, A. Belloum, and M. Bubak, “A framework for interactive parameter sweep applications,” in *Int Conf Computational Science (3)*, 2008, pp. 481–490.

[32] J. M. Brady *et al.*, “e-DiaMoND: A grid-enabled federated database of annotated mammograms,” in *Grid Computing: Making the Global Infrastructure a Reality*, ser. Willey, 2003, p. 923943.

[33] D. Rex, J. Ma, and A. Toga, “The LONI pipeline processing environment,” *NeuroImage*, vol. 19, pp. 1033–1048, 2003.

[34] N. Shadbolt *et al.*, “MIAKT: Combining grid and web services for collaborative medical decision making,” in *UK e-Science All Hands Meeting*, 2004.

[35] A. Rowland *et al.*, “Information extraction from images (IXI): Image processing workflows using a grid enabled image database,” in *DiDaMIC Workshop*, 2004.

[36] R. Buyya *et al.*, “Neuroscience instrumentation and distributed analysis of brain activity data: a case for eScience on global Grids,” *Concurrency and computation: practice and experience*, vol. 17, no. 15, pp. 1783 – 1798, 2005.

[37] S. Benkner *et al.*, “GEMSS: grid-infrastructure for medical service provision.” *Methods Inf Med*, vol. 44, no. 2, pp. 177–81, 2005.

[38] R. H. McClatchey, D. Manset, and A. E. Solomonides, “Lessons learned from mammogrid for integrated biomedical solutions,” in *19th IEEE Symp Computer-Based Medical Systems*, 2006, pp. 745–750.

[39] C. Barillot *et al.*, “ Federating Distributed and Heterogeneous Information Sources in Neuroimaging: The NeuroBase Project ,” in *Proceedings of HealthGrid 2006*. IOS Press, 2006, pp. 3–13.

[40] K. Nozaki *et al.*, “The first grid for the oral and maxillofacial region and its application for speech analysis,” *Methods Inf Med*, vol. 44, no. 2, pp. 253–6, 2005.

[41] T. C. Pan *et al.*, “GridCAD: Grid-based Computer-aided Detection System,” *RadioGraphics*, vol. 27, no. 3, pp. 889–897, 2007.

[42] J. Montagnat *et al.*, “NeuroLOG: a community-driven middleware design,” in *Proceedings of HealthGrid 2008*. IOS Press, 2008, pp. 49–58.

[43] S. Erberich *et al.*, “Globus MEDICUS - Federation of DICOM Medical Imaging Devices into Healthcare Grids,” in *Proceedings of HealthGrid 2007*. IOS Press, 2007, pp. 269–278.

[44] D. Krefting *et al.*, “MediGRID: Towards a user friendly secured grid infrastructure,” *Future Gener Comput Syst*, vol. 25, no. 3, pp. 326–336, 2009.

[45] S. G. Erberich *et al.*, “funcLAB/G service-oriented architecture for standards-based analysis of functional magnetic resonance imaging in HealthGrids,” *Neuroimage*, vol. 37, pp. S135–S143, 2008.

[46] M. Dojat and B. Gibaud, *Workshop on Distributed Databases and processing in Medical Image Computing (DiDaMIC)*, 2004. [Online]. Available: http://www.irisa.fr/visages/demo/Neurobase/Didamic/

[47] S. Olabarriaga, D. Lingrand, and J. Montagnat, *Workshop on Medical imaging on grids: achievements and perspectives*, 2008. [Online]. Available: http://www.i3s.unice.fr/~johan/MICCAI-Grid08/

[48] P. Kacsuk and G. Sipos, “Multi-Grid, Multi-User Workflows in the P-GRADE Grid Portal,” *J Grid Computing*, vol. 3, no. 3-4, pp. 221 – 238, 2005.

[49] H. Benoit-Cattin, F. Bellet, J. Montagnat, and C. Odet, “Magnetic Resonance Imaging (MRI) Simulation on a Grid Computing Architecture,” in *3rd IEEE Int Symp Cluster Computing and the Grid (Biogrid Workshop)*, 2003, pp. 582–587.

[50] J. Freund *et al.*, “Health-e-Child: An Integrated Biomedical Platform for Grid-Based Pediatrics,” in *Proceedings of HealthGrid 2006*. IOS Press, 2006, pp. 259–270.

[51] P. Wolniewicz *et al.*, “Accessing Grid computing resources with g-Eclipse platform,” *Comput Methods in Science and Technology*, vol. 13, no. 2, pp. 131–141, 2007.

[52] M. Kupczyk, R. Lichwa, N. Meyer, B. Palak, M. P, and P. Wolniewicz, “Applications on demand as the exploitation of the Migrating Desktop,” *Future Gener Comput Syst*, vol. 21, no. 1, pp. 37–44, Jan. 2005.