# From “low hanging” to “user ready”: Initial Steps into a HealthGrid

Sílvia D. OLABARRIAGA [a,b1], Tristan GLATARD [a,b], Kamel BOULEBIAR [a,b], Piter T. de BOER [b],

[a] *Academic Medical Center, University of Amsterdam, NL*

[b] *Institute of Informatics, University of Amsterdam, NL*

**Abstract.**
Grids offer powerful infrastructures and promising concepts for the development and deployment of advanced applications in medical research and healthcare. The construction of HealthGrids in practice, however, is challenging due to reasons of scientific, technical, and cultural nature, among them the large gap between communities that develop and use the technology. While grid developments focus mostly on functionality, usability issues are also very important to enable the potential of grids to be fully exploited by those who could mostly benefit from it, the end-users. In this paper we make a retrospective of our efforts to develop the Virtual Lab for functional Magnetic Resonance Imaging (fMRI). This project aims at providing for the end-users a grid-based system to facilitate research and clinical usage of fMRI data for study of brain activation. We present the evolution of this project in three phases coined “low hanging fruit”, “trying out” and “user-ready”, and the lessons learnt in each one. The evolution of the software architecture, which had a large impact on the user front-end, is discussed in more detail. The current software architecture facilitates the construction of front-ends that enable users to access the grid infrastructure from a single user-friendly GUI. All (local and grid) resources are accessed directly by the users from a virtual desktop implemented by a Virtual Resource Browser (VBrowser).

**Keywords.** functional MRI, medical imaging, grid front-end, grid usability, grid application, e-Science

## 1. Introduction

Grids have been appointed as a powerful infrastructure and promising concept for the development and deployment of advanced applications in medical research and healthcare. The construction of HealthGrids in practice, however, is challenging due to reasons of scientific, technical, and cultural nature. As a consequence, grid-enabled environments that are actually employed in the daily routine of medical researchers and practitioners are seldom found, in spite of their enhancing potential. One of the many challenges is the low usability of grid-based solutions, which are complex and still unstable systems that require much IT-related expertise. The end-users, which are potentially the most interested party, can hardly use these solutions directly.

In this paper we make a retrospective of our efforts to develop the Virtual Lab for functional Magnetic Resonance Imaging (fMRI [1]), a project that aims at building a system to facilitate and enhance research and clinical studies on brain activation due to stimulation. The target user community is the clinical end-users, therefore usability aspects are emphasized. Section 2 introduces the context and evolution of this project in three phases that we coined “low hanging fruit”, “trying out” and “user-ready”. The focus of this paper lies on the evolution of the software architecture (section 3), which had a large impact on the type of front-end available for the users. Section 4 concludes the paper briefly discussing related work and presenting current and future work.

## 2. Evolution of the Virtual Lab for fMRI

The Virtual Lab for functional MRI (VL-fMRI[2]) was initiated in 2005 in the scope of the Virtual Laboratory for e-Science Project [2]. It is motivated both by an “application pull” and a “technology push”. On the one hand, neuroscientists and clinical researchers at the Academic Medical Center of the University of Amsterdam (AMC) face growing problems to cope with the complexity and volume of data in their experiments (application pull). On the other hand, the Dutch e-Science infrastructure built by VL-e offers physical resources for data acquisition, storage and computing, as well as generic services that can be combined to solve domain-specific problems (technology push). The VL-fMRI explores this infrastructure to facilitate and enhance the study of brain activation with fMRI. Efforts are concentrated on the construction of user-friendly software that (could) enable end-users to directly benefit from the infrastructure without being hampered by its underlying complexity. The target community is composed of clinical users with background in Radiology, Psychology, Psychiatry, Medical Physics and Medical Imaging. The VL-fMRI has evolved through roughly three phases summarized below: initial (“low hanging fruit”), second (“trying out”), and current (“end-user ready”).

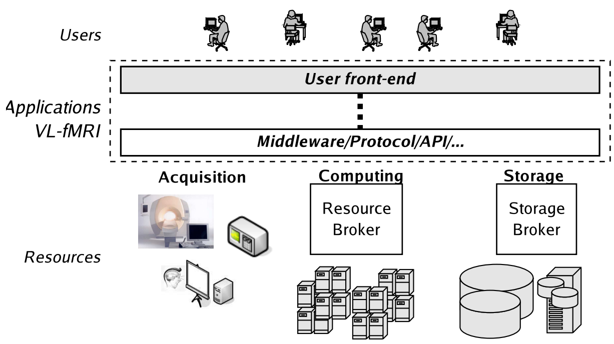


**Figure 1.** Components of the Virtual Lab for fMRI: users from distributed organizations, resources provided by the Dutch e-Science infrastructure, and grid-enabled applications.

[2]http://www.science.uva.nl/˜silvia/vlfmri

### 2.1. *Low hanging fruit*

This phase was coined “low hanging fruit” because the main idea was to build software by *simply* reusing components developed for other domains more advanced in grid usage, such as High Energy Physics (HEP). Initially we identified requirements, functional components, and proposed the basic architecture illustrated in Figure 1 – more details in [3]. Resources include image acquisition devices at the hospital, computing resources from the EGEE infrastructure[3] and data storage devices (San Diego Storage Resource Broker, SRB [4]). These resources are distributed among several organizations, being accessed via software tools composed of a front-end and several layers of software components and middleware. The VL-e project was taking shape, and so was the infrastructure, requirements, competences and team (`vlemed` VO). Limited hardware resources were available, and these could be accessed only via command-line utilities. Linux shell was the unique front-end to the gLite[4] and SRB middleware. Most of the effort was put into *(1)* establishing a multi-disciplinary team of Medical Imaging and Computer Science researchers, Grid developers and infrastructure providers, *(2)* identifying the application requirements and infrastructure constraints, and *(3)* dealing with failure and frustration.

The resulting implementation served to prove the concept: shell scripts were used to transfer data from the hospital to the SRB, perform image analysis on Grid resources, and transfer back results to the hospital for evaluation [3]. However, this implementation of the VL-fMRI was never accepted by the end-users; it was far too complex to use, the middleware was unstable, and the infrastructure provided insufficient computing resources. Moreover, the “batch usage model” promoted by the available middleware did not apply to the envisaged usage scenario, where users need to be (or at least feel) close to the data, being able to visually inspect images and results at any time. And finally, regulations on privacy simply forbid the usage of patient data in the proposed set-up. In fact, we found out that the fruit were still too high to catch - or our arms were simply too short. The most valuable results of this phase, however, consist of the multi-disciplinary team that grew out of it and the identification of the “gap” to be filled. In spite of being spread among many organizations, and clearly having different backgrounds, goals and priorities, the team members succeeded to develop a large variety of translational mechanisms for language, knowledge, and culture, consisting of a solid basis for further development.

### 2.2. *Trying out*

After realizing that “medical imaging is different from HEP”, as many had before us in the Healthgrid community [5], we set out for trying out several implementation alternatives with different front-ends and software stacks. The major goal was to improve the usability of the system, without sacrificing the generality of the components used to build it. As described below, many attempts used experimental set-ups outside the scope of the gLite middleware.

A significant improvement in the front-end for data resources was obtained with the Virtual Resource Browser (VBrowser) developed in VL-e to provide a more friendly desktop to the Grid [6]. This is an interactive tool that enables browsing (grid-enabled)

[3] *www.eu-egee.org*
[4] http://glite.web.cern.ch/glite/

resources from a single application. The user can access local and remote data with a familiar and intuitive look-and-feel, without being aware of the diversity of the protocols underneath (SRB, GridFTP, local file system, SFTP, etc.). Authentication with the various systems is performed on demand, as much as possible using GSI certificates for single sign-on. The VBrowser is fully written in Java, and as such compatible with Microsoft Windows, Linux and MacOS, providing users with a homogeneous interface from a variety of desktops. The interest shown by end-users increased significantly, since "access to the Grid no longer involved black windows"[5]. A few of them realized the potential beyond all the trouble, obtained grid certificates and installed the VBrowser on their desktops. With the Grid getting closer to those that really need it, a real demand started to appear for computing large scale fMRI analysis studies involving parameter and image sweeps.

To simplify the "gridification" of fMRI analysis jobs to run on the Grid, the fMRIB Software Library (FSL [7]) was pre-installed in all clusters accessible to the `vlemed` VO. Data were stored in the SRB server, being automatically staged-in/out the worker nodes by a wrapping script that also activates FSL `feat` to perform the complete image analysis pipeline for the given input data. Large experiments were performed in many ways, first based on shell scripts to submit jobs with gLite `edg-*` commands from a gLite-UI system. Later Python clients running on a regular desktop were used to access a custom WSRF-compliant grid service that interfaces with gLite middleware. Note that a similar web-service interface is currently adopted in the coming gLite WMS. The VLAM-G workflow management system [8] was also used to run fMRI analysis workflows, but at the time it was not ready for production grids, and could not be adopted in large scale. In particular, VLAM-G assumes resource reservation and in/outbound connectivity on the worker nodes, which are not realistic on the EGEE infrastructure. And finally, to facilitate monitoring and management of large numbers of jobs, we also adopted Nimrod [9]. Although Nimrod represented a significant improvement for running fMRI analysis experiments (see [10]), the set-up was still considered too difficult by the end-users because it involved diverse systems (yet another user account) and required considerable effort and programming skills to gridify and run new experiments. Moreover, at that time the Nimrod server at our disposal needed extra operation effort, since it did not fall under the services provided by the available gLite-oriented infrastructure. A grid expert was therefore still necessary to prepare, start and "babysit" experiments until the results were ready for the end-users.

Although there were (and still are) many other alternatives to be investigated, the lessons seemed clear to us. First, the computing facilities built in this phase were still too far from the end-users, requiring them to know too many technical details, log into (Linux-based) foreign systems, overcome firewall restrictions determined by local ICT administrators, deal with frequent and mysterious error conditions/messages, etc. To be successful, the VL-fMRI should bring a customized, but possibly less flexible, front-end to the user's desktop to provide access to services that encapsulate and hide the grid complexity both for computation and data access. Second, to really go through the effort of learning and adopting new tools, users need them to be reliable and performing, operated and supported on a "12/5" basis (at least). Such requirements are hard to reach with research software. In particular the stability requirement conflicts directly

[5] Quote from a user, referring to shell windows.

with the nature of the most interesting applications, which tend to adopt the newest techniques and prototypes that are insufficiently tested, debugged and supported. Moreover, we found that there is a large gap between experimental, controlled, grids (such as DAS-3 and Grid'5000) and the production infrastructures (such as EGEE), and scaling prototypes into usable applications still demands much effort. We therefore concluded that our team needed to operate in two fronts, by developing end-user ready applications suited for "production" Grids, in collaboration with grid infrastructure providers, but at the same time continuing to think ahead, collaborating with computer scientists to develop new/improved grid solutions and prototypes motivated by the requirements of our applications.

### *2.3. End-user ready*

A service-oriented architecture (SOA) approach was adopted to enhance system usability: (grid) services encapsulate and hide the complexity of accessing grid resources, and users contact these services via clients integrated into the VBrowser front-end. The first prototype addresses a simple and real use case where fMRI data is used for neurosurgery planning. The patient is scanned a few hours before the intervention for five different tasks: four activate motor areas (hands and feet individually) and one the language area. This case is considered *simple* because the scanning is short and image analysis is planned for maximum robustness and straightforward post-processing. The analysis is performed with pre-set parameters, takes $\pm$ 15 min per scan, and generates five brain activation maps. This case is *real* because it is part of a regular procedure adopted at the AMC in preparation to brain surgery for tumor extraction, using brain activation maps to locate vital areas to be spared. The analysis, however, sometimes "fails", i.e., the patient responds to the stimulation as expected, but no activation is visible in the brain activation map for one or more tasks. Currently a team of neuroradiologists and MR-physicists at the AMC are investigating improvements on the acquisition and analysis of fMRI data. Large parameter sweep studies are planned to find out the best configuration for this patient group. This study can only be performed using the VL-fMRI infrastructure, therefore it is important that end-users can directly prepare, run and monitor parameter sweep experiments on the Grid.

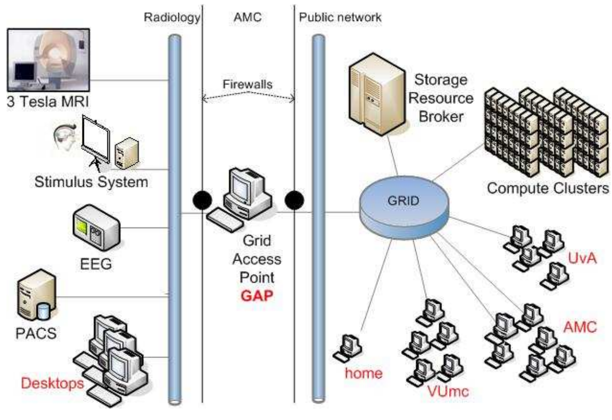


**Figure 2.** Systems inside the hospital are connected to external grid resources via the GAP.

Services are installed on a privileged system, coined Grid Access Point (GAP), which is connected to the AMC intranet and public network (Figure 2). The GAP has limited in/outbound connectivity with selected systems such as the scanning devices (radiology network), and the SRB, gLite Resource Broker (RB), Proxy and VOMS servers (public network). Due to security policy, it cannot be reached by regular users and systems from the public network. The GAP is configured as a gLite User Interface system that additionally runs a GT4 GridFTP server and other (custom) WSRF-compliant services that run on a secured GT4 container using HTTPS connection. Setting up this system was tremendously facilitated by the VL-e PoC[6] linux distribution, which includes packages such as gLite, GT4, FSL and others. In particular, this distribution made possible the use of a gLite UI without depending on CERN's Scientific Linux. Special firewalls configuration was facilitated by collaborative network administrators at the AMC.

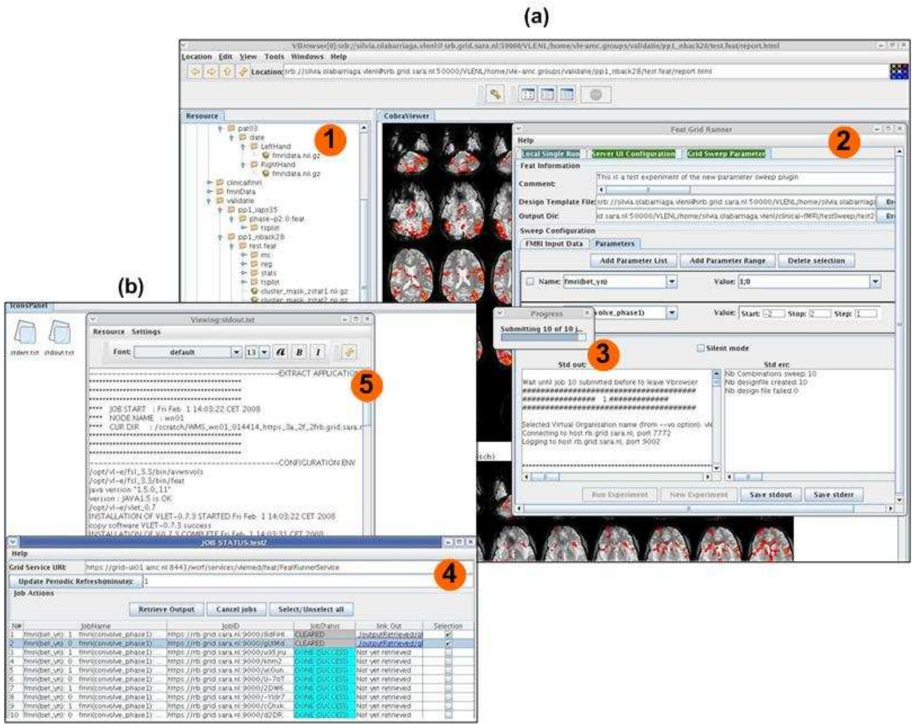


**Figure 3.** Screenshot of the feat sweep tool (a) and job monitoring tool (b). Specific interface elements are numbered: (1) resource tree (data and services); (2) experiment design, where the input data and parameters are chosen interactively; (3) status of job submission; (4) job monitoring and (5) job output.

The VBrowser was adopted as integrating platform due to its user-friendly interface, which has become familiar and accepted by the users, its extensibility via plug-ins that can be associated to MIME-types, and the existing support for development and debugging in VL-e. Two plug-ins provide the most relevant functionality for the considered use case. The "feat sweep tool" is associated with files that contain parameters for FSL `feat`

[6] *http://poc.vl-e.nl*

(*.fsf); see Figure 3-a. The user interactively selects input data (with a file browser that transparently accesses SRB or GridFTP servers), selects parameters to vary during the experiment (and ranges/lists of values), and indicates an output directory for the results. For each combination of input and parameter values, a job is submitted using gLite middleware. The job description language (jdl) file is generated automatically. To facilitate experiment management, the plug-in also saves metadata about the experiment in the output directory, such as all the parameter combinations, job identifiers, and links to the corresponding results. The "job monitoring tool" enables the user monitor the experiment status and retrieve the output of jobs interactively (see Figure 3). Results and metadata are stored in grid-enabled resources that can be viewed directly by the user from the VBrowser. The users contact and authenticate with the services using the VBrowser front-end from any desktop inside the hospital. Using the delegation service provided by the Globus container, the VBrowser and services transfer data and submit/monitor jobs on behalf of the user. The user certificate and private key remain on the local desktop.

The main characteristic of this phase has been excitement: the infrastructure (hardware and software) has been significantly improved, the team became larger and more experienced, and users became familiar with remote resources. The new system has been in operation for six months, but many improvements on the GUI were additionally required by the users. Differently from other phases, the VL-fMRI now seems to be ready for the end-users, being put into "production" as we write.

## 3. Looking Back: System Architectures

To accommodate the goals of VL-fMRI in the different phases, the system architecture has evolved as illustrated in Figure 4. The main features for each phase are summarized and compared in Table 1.

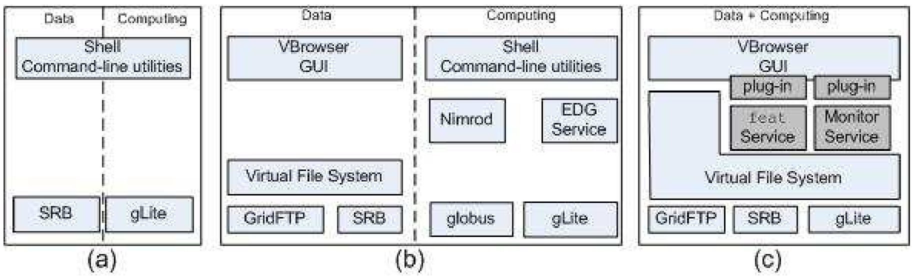


**Figure 4.** VL-fMRI software architecture in (a) first, (b) second, (c) current phases. Main components, top to bottom: front-end, specific and general services and middleware.

The initial architecture (Figure 4-a) was tightly coupled and required specific software (SRB commands, gLite) to be installed on the front-end system. Not only was the installation process complex, but also sometimes impossible because the users did not have sufficient rights. Access to the grid was done via command-line utilities (CLU) and users needed to create their proxy mastering the various flavors of this process (voms-proxy-init versus grid-proxy-init, VO specification, . . . ). Although this architecture seems far away for us now, it is still promoted as entry-level to new users that follow grid tutorials today.

| | Initial | Second | Current |
|---|---|---|---|
| User front-end | | | |
| Data access and transfers | SRB S* CLU | VBrowser GUI | VBrowser GUI |
| Job submission | edg* CLU | Python WSRF clients (CLU) | VBrowser plug-in (GUI) |
| Experiment description | scripts | Nimrod, scripts | VBrowser plug-in (GUI) |
| Software Requirements on user desktop | gLite UI | JRE (data), gLite UI Globus clients | JRE |
| Supported OS | Linux PoC | Linux PoC, Windows (data) | Linux (any), MacOS, Windows |
| Middleware | | | |
| Data | SRB | SRB, GridFTP | SRB, GridFTP |
| Job submission | gLite | gLite, Globus | gLite |
| Experiment (many jobs) | n.a | Nimrod server | GT4 app. service |

**Table 1.** Summary of system characteristics in different phases. Legend: CLU=command-line utilities, GUI=graphical user interface, JRE=Java Runtime Environment.

The second architecture (Figure 4-b) adopted two different approaches for data and computing resources. The `edg-*` commands were wrapped behind a grid service to facilitate submission from non-UI machines. The "Virtual File System" (VFS) was introduced to facilitate data management in the VBrowser. The VFS is part of the VL-e Toolkit (VLET[7]), which provides an abstraction layer to the middleware. JavaGAT [11] was also considered, however it did/does not have adaptors for the SRB or gLite. In this phase the system had completely different front-ends to access data and computing resources, requiring the user to constantly switch between look-and-feels and/or systems. This is not only less friendly, but also requires more management, for example, of user accounts, passwords, software and certificates. Note that this type of architecture is still the core in many applications, for example, when portals are used to run jobs, but the files still need to be manipulated by local applications that require the content to be externally down/uploaded.

In the current architecture (Figure 4-c), the SOA-oriented approach decouples the functionality and complexity (service) from the front-end (client), resulting in lighter software requirements on the user desktop. All tools (VBrowser, clients and services) are based on the VFS, such that data, metadata and results can be easily accessed by the used from a single and friendly GUI. The VBrowser, plug-ins, JRE and user certificates can be installed for example on a memory stick and activated from any Linux, Windows, or MacOS desktop.

## 4. Discussion and Final Remarks

We described the initial steps taken to implement a grid-enabled system to enhance the management and analysis of functional MRI data. From a simple and tightly coupled approach, the system evolved into a flexible service-oriented architecture following user

[7] *https://gforge.vl-e.nl/projects/vlet/*

requirements. Clients are integrated into a single front-end, provided by the VBrowser, from which the user can transfer files, view and edit remote data and metadata content, execute and monitor jobs and experiments, etc. The adopted architecture enables and facilitates the development of customized services and front-ends that have a higher chance of being adopted by end-user than the previous ones. An objective evaluation will be performed with the help of end-users in the near future.

Many other projects have devoted much attention to the development of user-friendly front-ends for grid-enabled applications, such as g-Eclipse [12], the Migrating Desktop [13], and the P-Grade portal [14]. In particular, the front-end of the Health e-Child project [15] aims at clinical users. Also the usage of grid-oriented approaches for analysis and management of medical imaging has become increasingly popular since 2004, as confirmed by the vast amount of existing literature, for example [16,17,18,19,20,21,22]. Although an objective comparison among the various alternatives is out of the scope of this paper, we feel that it would be important to determine in the future the fundamental differences among the various existing approaches, which ones are the most relevant, and under which conditions one approach represents an advantage over the other - or not. This would accelerate the development of new grid-enabled systems by helping motivated decisions about the software stack.

Motivated by the increasing interest of the end-users, we now focus on enhancing the system with additional components to handle medical image analysis workflows, security, and metadata management. For workflow management in large image analysis experiments on the grid, we will adopt MOTEUR [23], a workflow management system compatible with Taverna [24] and the EGEE infrastructure. For security, we are considering the Medical Data Manager [25] and Medicus [26], as well as collaborating with security experts [27]. Thanks to the flexibility of the current architecture, the new components will be integrated into the same familiar front of the VBrowser, such as presented in [28].

## Acknowledgment

We thank our brave and patient users A. Nederveen and S. Gevers, the VL-e colleagues who collaborate(d) in the design, development and operation of the VL-fMRI (J.J. Keijser, D. van Dok, M. Bouwhuis, B. Heupers, K.Maheshwari, J. Snel, R. Belleman, *grid.support@sara.nl grid.support@nikhef.nl*; the ICT group of the AMC (E.Wortel, M. Schrijnders, R.R. van Rijn); and the vision and support of Prof. dr. L.O. Hertzberger, Prof. dr. Grimbergen, Prof. dr. G. den Heeten, and Prof. dr. H. Lameris. The VL-e project is funded by a BSIK grant from the Dutch Ministry of Education, Culture and Science and part of the ICT innovation program of the Ministry of Economic Affairs.

## References

[1] S.M. Smith. Overview of fMRI analysis. *The British Journal of Radiology*, 77:S167–S175, 2004.

[2] L.O. Hertzberger. e-Science and the VL-e approach. In *Transactions in Computational Systems Biology IV*, volume 3939 of *LNCS*, pages 58–67. Springer Verlag, 2006.

[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 *Challenges and Opportunities of HealthGrids: Proceedings of HealthGrid 2006*, pages 43–54. IOS Press, 2006.

[4] C. Baru, R. Moore, A. Rajasekar, and M. Wan. The SDSC storage resource broker. In *CASCON'98 Conference*, 1998.
[5] V. Breton, K. Dean, and T. Solomonides. The Healthgrid White Paper. In *Proceedings of HealthGrid*, pages 249–321. IOS Press, 2005.
[6] S.D. Olabarriaga et al. Virtual Lab for fMRI: Bridging the Usability Gap. In *Proceedings of Second IEEE International Conference on e-Science and Grid Computing*, 2006.
[7] S.M. Smith et al. Advances in functional and structural MR image analysis and implementation as FSL. *NeuroImage*, 23:208–219, 2004.
[8] H. Afsarmanesh et al. VLAM-G: A grid-based virtual laboratory. *Scientific Programming*, 10(2):173–181, 2002.
[9] D. Abramson, R. Sosic, J. Giddy, and B. Hall. Nimrod: A tool for performing parametised simulations using distributed workstations. In *The 4th IEEE Symposium on High Performance Distributed Computing*, 1995.
[10] S.D. Olabarriaga, A. J. Nederveen, and B. Ó Nuallain. Parameter Sweeps for Functional MRI Research in the Virtual Laboratory for e-Science Project. In *Seventh IEEE International Symposium on Cluster Computing and the Grid (CCGrid 07)*, pages 685–690. IEEE Computer Society, May 2007.
[11] R. V. van Nieuwpoort, T. Kielmann, and H. E. Bal. User-friendly and reliable grid computing based on imperfect middleware. In *Proceedings of the ACM/IEEE Conference on Supercomputing (SC'07)*, 2007.
[12] P. Wolniewicz, N. Meyer, M. Stroinski, M. Stuempert, H. Kornmayer, M. Polak, and H. Gjermundrod. Accessing Grid computing resources with g-Eclipse platform. *Computational Methods in Science and Technology*, 2008. to appear.
[13] Miros Kupczyk, Rafa Lichwa, Norbert Meyer, Bartosz Palak, Marcin P, and Pawel Wolniewicz. Applications on demand as the exploitation of the Migrating Desktop. *Future Generation Computer Systems*, 21(1):37–44, January 2005.
[14] Péter Kacsuk and Gergely Sipos. Multi-Grid, Multi-User Workflows in the P-GRADE Grid Portal. *Journal of Grid Computing (JGC)*, 3(3-4):221 – 238, September 2005.
[15] J. Freund et al. Health-e-Child: An Integrated Biomedical Platform for Grid-Based Pediatrics. In *HealthGrid'2006*, volume 120 of *SHTI*, pages 259–270, Valencia, Spain, 2006.
[16] M. Pan and A. W. Toga. A grid enabled workflow management system for managing parameter sweep applications in neuroimaging research. In *Proceedings of Sixth IEEE International Symposium on Cluster Computing and the Grid Workshops*, 2006.
[17] J. Montagnat et al. Medical Images Simulation, Storage, and Processing on the European DataGrid Testbed. *Journal of Grid Computing*, 2:387–400, 2004.
[18] S. Kolbe et al. A Platform for Distributed Analysis of Neuroimaging Data on Global Grids. In *Proceedings of First IEEE International Conference on e-Science and Grid Computing*, 2005.
[19] V. Roberto et al. Neuroimaging Services on the Net. In *Proceedings of the 18th IEEE Symposium on Computer-Based Medical System*, 2005.
[20] S. Bagnasco et al. Early Diagnosis of Alzheimer's Disease Using a Grid Implementation of Statistical Parametric Mapping Analysis . In *Proceedings of HealthGrid*, pages 69–81. IOS Press, 2006.
[21] C. Barillot et al. Federating Distributed and Heterogeneous Information Sources in Neuroimaging: The NeuroBase Project . In *Proceedings of HealthGrid*, pages 3–13, 2006.
[22] D. Rogulin et al. A grid information infrastructure for medical image analysis. In *Proceedings of DiDaMIC Workshop (MICCAI'2004)*, 2004.
[23] T. Glatard, J. Montagnat, D. Lingrand, and X. Pennec. Flexible and efficient workflow deployement of data-intensive applications on grids with MOTEUR. *Intl. Journal of High Performance Computing and Applications (IJHPCA)*, 2008. to appear.
[24] T. Oinn et al. Taverna: A tool for the composition and enactment of bioinformatics workflows. *Bioinformatics Journal*, 10(17):3045–3054, 2004.
[25] J. Montagnat et al. A Secure Grid Medical Data Manager Interfaced to the gLite Middleware. *Journal of Grid Computing (JGC)*, 2007.
[26] S. G. Erberich et al. Globus MEDICUS - Federation of DICOM Medical Imaging Devices into Healthcare Grids. In *Proceedings of HealthGrid*, volume 126 of *Studies in Health Technology and Informatics*, pages 269 – 278, 2007.
[27] G. J. van Noordende, S. D. Olabarriaga, M. R. Koot, and C. Th.A.M. de Laat. Privacy and trust for grid-based medical applications. submitted.
[28] T. Glatard, K. Boulebiar, and S. D. Olabarriaga. Workflow integration in VL -e medical. submitted.