Towards Production-level Cardiac Image Analysis with Grids

Ketan Maheshwari, Tristan Glatard, Joel Schaerer, Bertrand Delhay, Sorina Camarasu-Pop, Patrick Clarysse, Johan Montagnat

Towards Production-level Cardiac Image Analysis with Grids

Ketan MAHESHWARI [a,1], Tristan GLATARD [b], Joël SCHAERER [b], Bertrand DELHAY [b], Sorina CAMARASU-POP [b], Patrick CLARYSSE [b] and Johan MONTAGNAT [a]

[a] *University of Nice – Sophia Antipolis / CNRS, I3S laboratory, 06903 Sophia Antipolis, France.*

[b] *University Lyon 1, CREATIS-LRMN; CNRS UMR5220; Inserm U630; INSA-Lyon; France.*

Abstract. Production exploitation of cardiac image analysis tools is hampered by the lack of proper IT infrastructure in health institutions, the non trivial integration of heterogeneous codes in coherent analysis procedures, and the need to achieve complete automation of these methods. HealthGrids are promising technologies to address these difficulties. This paper details how they can be complemented by high level problem solving environments such as workflow managers to improve the performance of applications both in terms of execution time and robustness of results. Two of the most important important cardiac image analysis tasks are considered, namely myocardium segmentation and motion estimation in a 4D sequence. Results are shown on the corresponding pipelines, using two different execution environments on the EGEE grid production infrastructure.

Keywords. Cardiac image analysis, workflow enactment, grid computing, production.

1. Production-level Cardiac Image Sequences Analysis

Cardiac diseases are one of the major causes of death in industrial countries. Cardiac diagnosis is increasingly relying on temporal sequences of 3D images acquired using fast Magnetic Resonance Imaging (MRI) sequences. Such sequences, routinely acquired in hospitals, represent wealth of imaging data which require processing to extract meaningful quantitative parameters useful for diagnosis. Cardiac image analysis procedures are applied to analyze the patient heart status, through *e.g.* studies of the heart shape (myocardium segmentation) and heart motion (myocardial motion estimation).

Although analysis methods are actively developed and tested in research, image sequences are still often exploited manually in clinics, despite the extremely tedious work involved, such as the delineation of the myocardial contours in complete time series of MR slice stacks. Usage of analysis tools in production is hampered by several factors including (**F1**) the integration of heterogeneous codes to conduct the analysis, (**F2**) the need to fine-tune analysis parameters to adapt to the variability among images and (**F3**)

[1] Corresponding Author: Ketan Maheshwari, http://www.i3s.unice.fr/~ketan

the associated computing and data storage requirements when dealing with large image databases.

HealthGrids are appealing to address those challenges. Concerning (**F1**), workflow technology can be considered. Pipeline-based image processing tools with a visual programming interface, such as the commercial Khoros and AVS tools, have been widely used by non-expert end users. In the medical image analysis community, the ITK[2] open source library has become a *de facto* standard providing a pipeline design API. In our work, we study workflow technology to provide similar functionality in a HealthGrid production environment. Unlike Khoros or AVS, the workflow managers are intrinsically parallel execution engines able to outsource computations on distributed units, as well as exploiting local resources. In addition, the model adopted for pipelines description is a loosely coupled workflow of application services. Unlike ITK, this does not require code integration and compilation in a single unit. Heterogeneous application codes are embarked in the workflows, thus maximizing flexibility and providing easy reconfiguration for pipeline prototyping and parameter sweep studies set up.

Regarding (**F2**), complete automation is a complex problem due to the difficulty to automatically tune algorithms to adapt to various images and the need for expert validation. Parameter-sweeps can be considered to cope with this issue. In HealthGrids, they already have been used to perform validation (e.g, in [8] for functional MRI) or large-scale population studies for instance in neuroimaging [10]. Such experiments are naturally expressed in workflow languages and executed in a distributed infrastructure with engines like Taverna [7], MOTEUR [4] or Nimrod-G [15]. Wrapper tools such as GASW [3] or Soaplab [14] facilitate the integration of tools in those environments.

Finally, (**F3**) can be addressed by leveraging the resources made available by production grids like EGEE[3] which provides a computational back-end that is often lacking in clinical centers to support secured and data intensive applications. However, those platforms also provide high latencies and low reliability as a downside of the computing power and storage capacity. For instance, [1] reports a 55% error ratio on a metagenomics application. In this context, exploiting local computing resources for part of the experiment can yield better performance.

In this paper we study the ability of HealthGrids to support production-level cardiac image analysis. First, pipelines implementing myocardium segmentation and cardiac motion estimation are described. Then, parameter-sweep experiments are shown on the segmentation pipeline to tune initialization and deformation parameters with the aim to go towards an automatic method. Finally, a performance comparison of the motion estimation pipeline running on EGEE versus local resources is presented, evaluating the trade-off between grid and local execution.

2. Cardiac Image Sequence Analysis Pipelines

Two cardiac analysis pipelines developed in the context of the GWENDIA[4] project are considered for experimentation (see figure 1). The myocardium *segmentation pipeline* is

[2] Insight ToolKit, http://www.kitware.com/

[3] Enabling Grids for E-sciencE, http://www.eu-egee.org

[4] Grid Workflows Enactment of Data Intensive Applications project, http://gwendia.polytech.unice.fr

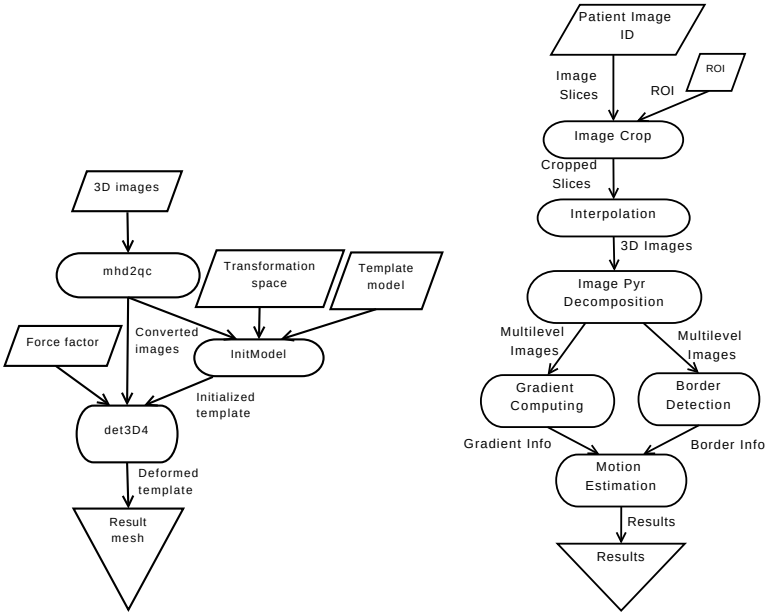


Figure 1. Myocardium segmentation (left) and motion estimation (right) pipelines.

an elastic template-based left and right myocardium extraction procedure. Segmentation is a process sensitive to initial conditions and which needs to be tuned to take into account the patient specific image properties. The heart *Motion Estimation pipeline* is a multi-step workflow extracting a dense vector field for each time instant of the sequence which estimates the heart structure motion between successive image pairs. This pipeline features a computer intensive algorithm and is used for performance tuning. Both pipelines are designed to process the same multislice and multiphase Cine MRI databases.

2.1. Myocardial Segmentation

To achieve automated batch processing of patient data, the segmentation pipeline steps should require as little human intervention as possible. Segmentation traditionally involves manual steps for initialization and results checking. Grid technology can help by providing workflows management tools to perform sweeps on application parameters as exemplified in [11] for brain segmentation.

In this study, we consider the myocardium segmentation pipeline represented in figure 1 (left), composed of three steps:

- `mhd2qc` is a conversion step working on 3D volumes produced by the interpolation step;
- `InitModel` is the initialization, during which an *a priori* heart model is positioned close to the heart using affine transform;
- `det3D4` is the segmentation step itself, which deforms the model to fit the myocardial borders.

The two last steps usually require human intervention: initialization is performed under user control by interactively positioning the model in the 3D space close to the heart. The correctness of the segmentation is visually checked and validated by a physician.

In addition to the data itself (3D images), this workflow requires a volumetric prior template mesh model representing the geometry of the heart interfaces and cavities. This template also carries mechanical properties of the heart tissues. It is first placed within the image close to the structure to be extracted. To achieve an automated procedure for initialization, a random search was implemented. A transformation space (i.e. min/max values for the 7 parameters describing translation and rotation in 3D as well as isotropic scaling) is randomly browsed, computing a closeness measure (energy) between the model and the myocardium for each candidate transformation. Parameters leading to the best energy are kept to initialize the template model for the segmentation step.

The segmentation algorithm is based on a Deformable Elastic Template [12,13]. The template mesh is deformed iteratively by applying a force field so that its edges stick to the borders of the targeted structure. Gradient and edge information is first extracted from the 3D images composing the cardiac image sequence. Edge maps are derived into Gradient Vector Fields that guide the deformation process through a classical energy minimization procedure. The force field is weighted w.r.t the template prior with a parameter coined *force factor*: the higher the force factor, the higher influence the data has during the deformation process. To improve convergence speed and accuracy, a multi-resolution approach is implemented for both the model and the image.

2.2. Cardiac Motion Estimation

The Motion Estimation pipeline is graphically represented in Figure 1 (right). The first step is a cropping applied to a series of DICOM slices identified by a patient ID. A Region Of Interest (ROI) is used to crop all the images around the heart region. Next, 3D images are assembled and the data set is isotropically interpolated. To speed-up and improve robustness of the Motion Estimation process, the isotropic images are undersampled into a multi-resolution pyramid. Gradient and edges are then extracted at each resolution level. Finally, the motion inside the region of interest is estimated throughout the sequence based on non rigid registrations between consecutive image pairs (free form deformations with the sum of squared difference as similarity measure) [2]. Motion estimation is a compute intensive process which execution time is unacceptable when dealing with large data sets.

2.3. Workflow Managers

Workflows presented above can be executed by MOTEUR and Taverna. In these frameworks, a workflow is described as a collection of processors connected by data links, which establish a dependency between the output(s) of a processor and the input(s) of a subsequent one. The engine orchestrates the execution of the processors in a way that is consistent with the dependencies. It manages the flow of data through the processors. The execution is driven by a push model: a processor's execution is started as soon as all of its inputs are available. It is important to note that the pipelines pictured in figure 1 are designed for processing multiple patient image databases. Computation tasks are generated for all processing steps and all images considered in the experiments. The execu-

tion of the workflows is optimized by exploiting coarse level parallelism: independent computing tasks and independent data sets can be processed concurrently.

MOTEUR. MOTEUR[5] is a data-intensive workflow manager interfaced to EGEE through a Grid Application Service Wrapper (GASW). It was designed for maximal performance and it exploits three levels of parallelism that are implicitly expressed in the workflow and its data sets. First, MOTEUR concurrently enacts independent branches of the workflow. Second, it performs pipelining: the workflow stages are pipelined with different data items to be processed. Third, MOTEUR exploits data parallelism: a same stage is concurrently enacted for all independent data items to be processed.

Taverna. Taverna[6] version 2, is a dataflow manager developed by the ^{my}Grid consortium. It includes a GUI-based rich-client workbench for workflow design and a workflow enactment engine. The default enactment mode of Taverna 2 is pipelining. In addition Taverna 2 provides a so called "super-pipelining" execution mode that corresponds to MOTEUR's data parallelism. Taverna 2 supports various processor types invocation, including local execution and Web Services. The core of Taverna 2 was extended through an EGEE plugin that automates grid jobs submissions and data transfers to the grid nodes.

2.4. Data

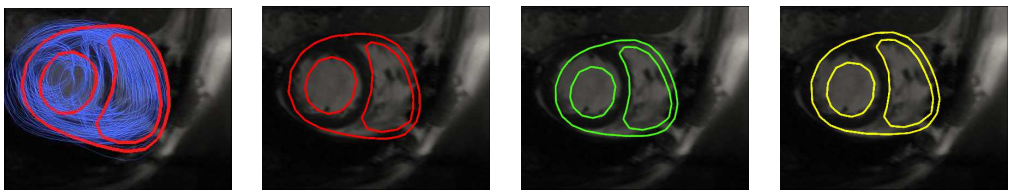
A set of 10 MRI examinations on patients has been collected for this study. Cardiac cine MR images were acquired with a 1.5 T magnetic resonance Sonata scanner (Siemens, Erlangen) at the Cardiology Hospital of Lyon, France. An ECG-gated True-Fisp sequence was used to acquire 5 to 11 short-axis cine slices covering the heart with 208-240$\times$256 pixel matrix, 5-7 mm slice thickness, 10-12 mm slice-spacing. Each sequence was composed of 11 to 30 frames over the cardiac cycle.

3. Parameter-sweeps for Segmentation Automation

3.1. Infrastructure setup

Experiments described in this section were performed using the MOTEUR workflow manager for execution [4] and the VBrowser interface for data management [9]. The EGEE Logical File Catalog (LFC) was used for data management, all the files being stored on a single Storage Element (SE). To ease prototyping for those early experiments, jobs were submitted on a local server featuring an Intel Xeon 5410 (2.33 GHz quad core) that also hosted the workflow manager. We dedicated 3 of the 4 cores to computing jobs and 1 to the workflow manager. The server ran an SL4 EGEE User Interface with no pre-installed software (dependencies were all downloaded from the SE). Jobs were thus running in similar conditions than on EGEE. Scaling those experiments to the grid just requires to switch a flag in the setup configuration.

[5] MOTEUR workflow manager: http://modalis.i3s.unice.fr/softwares/moteur/start
[6] Taverna workflow manager: http://taverna.sf.net



(a) (b) 100 / 150 / 20°/ 2.1 (c) 100 / 200 / 20°/ 1.9 (d) 100 / 150 / 10°/ 2.1

Figure 2. Parameter sweep on initialization parameters. (a) 75 parameter have been tested ; (b),(c) and (d) initialization results kept for segmentation. Parameter values for Txmin / Txmax / Rzmax / Smin are shown. Image slices are in an `xy` plane.

3.2. Experiments and results

Using this setup, two parameter-sweep experiments have been conducted to study automatic segmentation of the database. First, the bounds of the transformation space to use for initialization were investigated. Second the influence of the force factor in the deformation process was looked into.

Experiment #1. Bounds of the transformation space are defined by 14 parameters, namely the minimal and maximal values of the translation vector coordinates, the rotation angles and the scaling factor. Early tests showed that the bounds of the translation along x (`Txmin` and `Txmax`), the maximal rotation angle w.r.t z-axis (`Rzmax`) and the min of the scaling factor (`Smin`) had significant importance on the quality of the resulting initialization. Consequently, in this experiment, `Txmin` was varied among 3 values, `Txmax` and `Rzmax` among 2 and `Smin` among 6 for one volume of a patient, leading to 72 initializations. Figure 2-(a) shows in blue the 72 computed initializations. Among those, a couple only could be considered correct from a visual check. We kept the red, green and yellow results shown in (b), (c) and (d) for the segmentation step. Execution time statistics are reported in table 1 for the `InitModel` workflow processor.

Experiment #2. The force factor was varied from 0.1 to 0.5 by steps of 0.1. For each initialization, 5 segmentations were thus performed. Due to issues with workflow iteration strategies, this experiment had to be performed independently from the previous one, running the workflow once for each initialization selected for segmentation. Figure 3 shows segmentation results obtained from selected initialization results. Colors denote the force factor value. Table 1 shows the corresponding execution statistics for the `det3D4` workflow processor.

3.3. Discussion

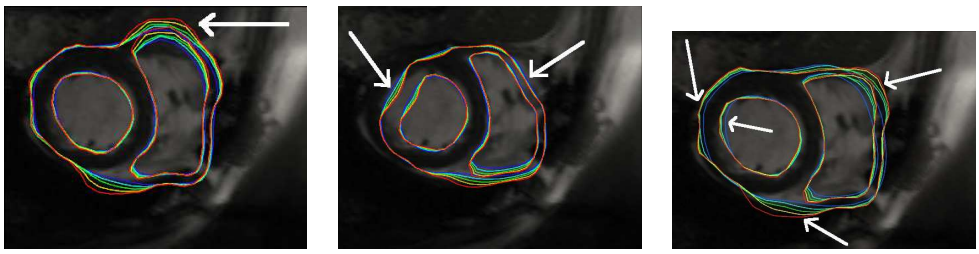
Experiment #1. Figure 2 demonstrates a significant influence of the swept parameters on the initialization results. As shown on part (a) of the figure (blue edge lines), the set of initialization results is quite widely spread. Among those, a couple of results can be considered as correct initializations. The selected ones for this study ((b), (c) and (d) part of the figure) have subtle but visually noticeable differences. Execution times reported on table 1 show a significant importance of data transfers, representing more than 50%

	#jobs	workflow step (see Fig. 1)	Elapsed	CPU (total ; $\mu \pm \sigma$)	Download (total ; $\mu \pm \sigma$)	Upload (total ; $\mu \pm \sigma$)
Exp.#1	72	`InitModel`	2h19'	3h53' ; 3'14"±59"	1h53' ; 1'34"±29"	37'42" ; 31"±4"
Exp.#2 - (b)	5	`det3D4`	*39'16"*	32'2" ; 6'24" ±5"	2'50" ; 34"± 1"	59" ; 12" ± 1"
- (c)	5	`det3D4`	*59'5"*	32'21" ; 6'28" ± 4"	2'48" ; 33" ± 1"	58" ; 12" ± 1"
- (d)	5	`det3D4`	*16'41"*	32'41" ; 6'32"±13"	2'57" ; 35"±2"	59" ; 12"±1"

Table 1. Execution statistics for the segmentation experiments. Mean (resp. standard-deviation) is denoted by μ (resp. σ). Data transfers are performed between our local server and an EGEE Storage Element. Italicized elapsed time values have been measured in shared conditions.

of the CPU time. Still, a fair speed-up of about 2 (on 3 processors) is obtained comparing the elapsed time to the total CPU time. In the future, data transfer optimizations specific to parameter sweeps could be envisaged since most of the jobs share the same set of input files.

Experiment #2. Although differences among initializations (b), (c) and (d) are quite subtle, segmentation results significantly differ, (d) being the only acceptable result from a visual point of view (see figure 3). Segmentation of the left ventricle is acceptable on (b) but the right ventricle is overestimated (see white arrow). Result (c) clearly underestimates the myocardium, both in left and right ventricles. The force factor also has a strong influence on the quality of the results. On (d), a high force factor (red edge) better segment the left ventricle (leftmost arrows) whereas a small one (blue edge) better fits the right ventricle contours (rightmost arrows). On this slice, a force factor of 0.2 (cyan edges) seems to be the optimal. Execution times reported on table 1 show a better trade-off between CPU and transfer times, the latter representing an eighth of the former. Job times are still short though compared to usual latency values on EGEE. To execute this workflow on EGEE, latency reduction solutions such as pilot-jobs [6] will be considered.



(b) (c) (d)

Figure 3. Parameter sweep on segmentation parameters. Five force factors have been tested for each initialization, as denoted by the color scale (blue: 0.1 ; cyan: 0.2 ; green: 0.3 ; yellow: 0.4 ; red: 0.5). Letters refer to figure 2.

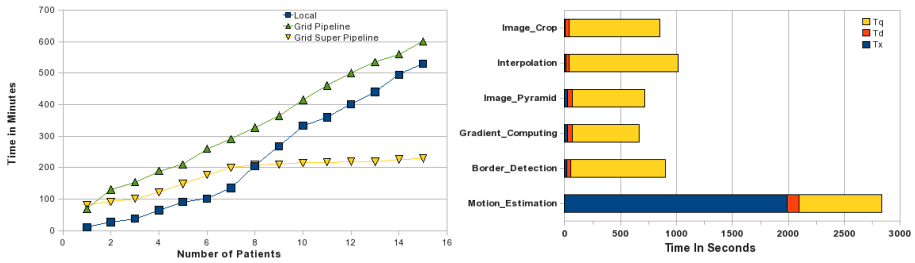


Figure 4. Cardiac Motion Estimation Results. Left: makespan when processing an increasing number of data. Right: average grid profile of each pipeline stage.

4. Performance Study for Motion Estimation

4.1. Infrastructure setup

Experiments presented in this section have been executed on EGEE, using Taverna 2 to orchestrate the workflow execution. EGEE is a premiere grid infrastructure federating 250 computing centers and dedicated to scientific production. Our experiments exploit the significant share of the infrastructure ($\approx$ 20000 CPU cores) accessible to the "biomed" Virtual Organization. In the context of our experiments, EGEE provides the computing resources that are usually not available for processing inside clinical centers. However, the grid is a batch infrastructure that introduces submission overheads in the tasks execution time. The tasks submitted and the data they depend on have to be transported to remote grid nodes. In addition, tasks are queued in multi-user queues and may experience unbounded delay, depending on the uncontrollable workload experienced by the infrastructure at the time of submission.

Furthermore, such a large scale infrastructure exhibits low reliability and job frequently fail due to network, system or middleware faults. During workflow execution, the frequency of job failures are magnified with the increased number of tasks, lowering the probability to succeed in executing a complete workflow. The following faut recovery measures were implemented in Taverna's EGEE plugin in order to increase its robustness: (i) a job resubmission policy (after a certain waiting time the jobs are resubmitted [5]), (ii) a Round-Robin selection policy for EGEE Workload Manager Services, and (iii) resubmitting the data transfer requests in case of failures.

4.2. Results

The left graph of figure 4 presents the scalability results for the Motion Estimation pipeline executed on the EGEE grid through the Taverna 2 plugin. The blue "local" curve corresponds to a sequential execution on a mono-processor local machine, used as a baseline. The entire workflow is submitted for an increasing number of patients (hence an increasing number of image sequences) and the makespan of the application is measured. It can be seen that the sequential execution scales with the number of patients. The application behavior is not completely linear though. The optimization algorithm used for motion estimation has a non-guaranteed execution time (the number of iterations depends on the data content), and therefore each patient processing time differs slightly.

The green "pipeline" curve reports the grid execution makespan using a Taverna regular pipelined execution. Unexpectedly, grid execution does not improve performance in this case, even when processing the complete data base. This result is expected when processing few data: a production grid overhead is non negligible as illustrated in the right graph of figure 4 which shows the average grid profile of main processes involved in the workflow. The tasks execution time *Tx* is impacted by the additional overhead composed of *Td*, the data transfer time from grid storage to the computing units, and *Tq*, the time spent inside the job queue after the job has been submitted and before the start of its execution. However, the overhead is expected to be compensated by parallel execution for a sufficiently large database processing. This does not happen in pipelined execution for this experiment because of the unbalanced computation time of the pipelines stages: the motion estimation execution time (Tx) largely dominates in this workflow and it is worsen by a high data transfer time (Td) which owes to the growth of data through the pipeline (around 0.5 GB from around 1 MB on an average). In pipelined mode, the multiple instances of motion estimation (one per patient) are not parallelized. The parallel gain is limited by the concurrent execution of motion estimation and other pipeline stages, thus leading to a high makespan.

Finally, the "super pipeline" curve shows the performance when enabling Taverna 2 "superscalar" pipeline submission.The concurrent execution of all workflow processes significantly increases overall application performance. The application performance is then mostly impacted by grid overhead. As soon as two or more patient datasets are processed, performance improves as compared to regular pipelining. Finally it beats the local execution after the 8th patient is processed. After this "cut-off" stage the performance dramatically improves as the concurrently executing processes start returning results almost simultaneously.

5. Conclusions

Production-level cardiac sequences analysis is made difficult by the lack of proper IT infrastructure in health institutions, the non trivial integration of heterogeneous components in tools analysis procedures, and the need to achieve complete automation. Today, HealhGrids provide a production stable environment serving as a foundation for tackling the higher level difficulties. In this paper we complement the grid infrastructure with high level workflow enactment tools that help to interface heterogeneous codes and benefit from grid performances. Alternative solutions exists such as MOTEUR and Taverna workflow managers, both interfaced to the EGEE infrastructures. Care should be taken in the parameterization of such tools though, as can be seen through the inefficient performance of the motion estimation experiment in simple pipelined mode.

Parameter sweep appears to be a good way to improve the robustness of the segmentation pipeline, for the initialization step as well as for the deformation. A few other parameters of the segmentation algorithm could be similarly be considered for sweeping. Using grids to support this kind of experiment is promising to go towards an automation of the segmentation pipeline required for mass image processing. From a performance point of view, pilot-job frameworks will be exploited in this pipeline since segmentation tasks are still short w.r.t the average grid latency. Data transfer optimisation strategies will also be considered since many jobs have the same set of input files in this kind of parameter-sweep experiment.

Grid computing power and flexible pipelines management capability are a promising combination to reach production-level cardiac image analysis with HealthGrids.

Acknowledgments. This work is supported by the French ANR GWENDIA under contract number ANR-06-MDCA-009. The Région Rhônes-Alpes is gratefully acknowledged for its support through the project Simed of the cluster ISLE. We are grateful to MD-PhD P. Croisille for providing the MR examinations exploited in this study.

References

[1] Gabriel Aparicio, Ignacio Blanquer Espert, and Vicente Hernández García. A Highly Optimized Grid Deployment: the Metagenomic Analysis Example. In *Global Healthgrid: e-Science Meets Biomedical INformatics (Healthgrid'08)*, pages 105 –115, Chicago, USA, May 2008. Healthgrid, IOS Press.

[2] B. Delhay, P. Clarysse, J. Lötjönen, T. Katila, and I. E. Magnin. Evaluation of two free form deformation based motion estimators in cardiac and chest imaging. In *Functional Imaging and Modeling of the Heart (FIMH)*, volume 3504, pages 467–476, Barcelona, Spain, 2005. Springer.

[3] Tristan Glatard, Johan Montagnat, David Emsellem, and Diane Lingrand. A Service-Oriented Architecture enabling dynamic services grouping for optimizing distributed workflows execution. *Future Generation Computer Systems*, 24(7):720?–730, July 2008.

[4] Tristan Glatard, Johan Montagnat, Diane Lingrand, and Xavier Pennec. Flexible and efficient workflow deployement of data-intensive applications on grids with MOTEUR. *International Journal of High Performance Computing and Applications (IJHPCA)*, 22(3):347–360, August 2008.

[5] Diane Lingrand, Johan Montagnat, and Tristan Glatard. Estimating the execution context for refining submission strategies on production grids. In *Assessing Models of Networks and Distributed Computing Platforms (ASSESS / ModernBio) (CCgrid'08)*, pages 753 – 758, Lyon, May 2008. IEEE.

[6] Jakub T. Mo. Distributed analysis environment for HEP and interdisciplinary applications. *Nuclear Instruments and Methods in Physics Research A*, 502:426–429, 2003.

[7] Tom Oinn, Matthew Addis, Justin Ferris, Darren Marvin, Martin Senger, Mark Greenwood, Tim Carver, Kevin Glover, Matthew R. Pocock, Anil Wipat, and Peter Li. Taverna: A tool for the composition and enactment of bioinformatics workflows. *Bioinformatics journal*, 17(20):3045–3054, 2004.

[8] Silvia Olabarriaga, Piter T. de Boer, Ketan Maheshwari, Adam Belloum, Jeroen Snel, Aart Nederveen, and Maurice Bouwhuis, editors. *Virtual Lab for fMRI: Bridging the Usability Gap*, Amsterdam, December 2006. IEEE.

[9] Silvia Olabarriaga, Tristan Glatard, Kamel Boulebiar, and Piter T. de Boer. From 'low-hanging' to 'user-ready': initial steps into a healthgrid. In *HealthGrid'08*, pages 70–79, Chicago, June 2008.

[10] Michael J. Pan and Arthur W. Toga. A grid enabled workflow management system for managing parameter sweep applications in neuroimaging research. In *Fourth International Workshop on Biomedical Computations on the Grid (Biogrid'06)*, Singapore, May 2006.

[11] Erik Pernod, Jean-Christophe Souplet, Javier Rojas Balderrama, Diane Lingrand, and Xavier Pennec. Multiple Sclerosis Brain MRI Segmentation Workflow deployment on the EGEE Grid. In Silvia Olabarriaga, Diane Lingrand, and Johan Montagnat, editors, *MICCAI-Grid Workshop (MICCAI-Grid)*, New York, NY, USA, September 2008.

[12] Q.-C. Pham, F. Vincent, P. Clarysse, P. Croisille, and I. E. Magnin. A FEM-based deformable model for the 3D segmentation and tracking of the heart in cardiac MRI. In *2nd International Symposium on Image and Signal Processing and Analysis (ISPA 2001)*, volume 1, pages 250–254, Pula, Croatia, 2001.

[13] Y. Rouchdy, J. Pousin, J. Schaerer, and Patrick Clarysse. A nonlinear elastic deformable template for soft structure segmentation: application to the heart segmentation in MRI. *Inverse Problems,*, 23:1017–1035, 2007.

[14] Martin Senger, Peter Rice, and Tom Oinn. Soaplab - a unified Sesame door to analysis tool. In *UK e-Science All Hands Meeting*, pages 509–513, Nottingham, September 2003.

[15] Wibke Sudholt, Kim K. Baldridge, David Abramson, Colin Enticott, Slavisa Garic, Chris Kondric, and Duy Nguyen. Application of grid computing to parameter sweeps and optimizations in molecular modeling. *Future Generation Computer Systems*, 21(1):27–35, 2005.