ASCLEPIOS

INRIA – Sophia Antipolis

INTERNSHIP

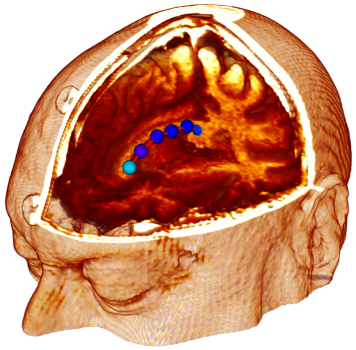
to fulfill the requirements for the degree of

Engineer in Numerical modelling

from the

Institut des **S**ciences de l'**I**ngenieur de **T**oulon et du **V**ar

Interfaces Development for medical imaging applications in the framework of the Multiple Scleoris disease



Erik PERNOD

Internship supervized
by
Jean-Christophe SOUPLET
March – August, 2008

Aknowledgments

First, I would like to express my gratitude to Dr Ayache who host me in the Asclepios team. then I would like to express profond gratitude to my advisor, Jean-Chritophe Souplet, whose support, encouragement and knowledge, added considerably to my graduate experience. I am also highly thankful to Xavier Pennec for giving me the chance to take a sizeable part in the NeuroLOG project.

I should not forget to express my immense gratitude to Javier Rojas Balderrama, Diane Lingrand and Johan Montagnat from the I3S laboratory, for the time they took to help me for the work on the grid. To Grégoire Malandain for presenting my article to the MICCAI 2008 grid workshop in New York.

I would like also to thank Nicolas Toussaint for the time he took helping me in programming and to be also a friendly labmates as well as François Hebert, Ender Konukoglu, Jean-Marc Peyrat and to the rest of the team who make this internship more pleasant.

Finally, I thank Rebecca to lending me her coffee maker which allows me to save a lot of money.

This internship has been supported by the NeuroLOG project.
ANR contract number: ANR-06-TLOG-024.

Abstract

Image processing has become an essential component in many fields like clinical practice and bio-medical research. It provides to clinicians helpfull tools which could confort their diagnosis. This is the reason why, nowadays, there is a demand of development of medical image processing and visualization tools.

The segmentation of lesions on brain MRI is required for diagnosis purpose in Multiple Sclerosis (MS). Moreover, the lesion burden is also used in MS patients' follow-up and in MS clinical researches. To this purpose, in our pipeline, the segmentation and the characterization of healthy tissues in multi-spectral MRI is the first step in order to separate them from lesions. However, automatic brain MRI segmentations methods are computationally intensive tools in medical image computing. Deploying them on grid infrastructures can provide an efficient resource for data handling and computing power.

In this study, an efficient implementation of a brain MRI segmentation method through the grid-interfaced workflow enactor MOTEUR on the EGEE grid is proposed. This work is part of the NeuroLOG project which aims at federating medical data and algorithms, and sharing computing resources on the grid.

A description of the tools and the steps needed to transform this kind of medical imaging application in a workflow is provided. Keeping in mind that a full knowledge of the pipeline is needed in order no to generate processing error. To become familiar with this last one, implementation of new functionality in the software SepINRIA have been done. This software is dedicated to Multiple Sclerosis brain MRI and is based on the same algorithms.

Besides, We particularly attach importance to the possibility and the interest of parallelism allowed by a workflow structure. As different kind of parallelism are available, we underline the fact that strategies of iterations have to be define to well structure the workflow. These strategies are done throw operators which combine inputs and outputs of the different services composing the workflow.

Results obtained from the grid have been validated by comparing them to the results compute locally on only one computer. Thus, time performance and description of the services made available to the NeuroLOG community are proposed. Finally, thanks to the power of the grid, method's parameter influence on the resulting segmentations is assessed given the best compromise between algorithm speed and results accuracy. This deployment highlights also the grid issue of a bottleneck effect.

Resumé

Le traitement d'image est devenue une composante essentielle dans de nombreux domaines comme en médecine ou en recherche bio-médicale. ce procédé fournit aux cliniciens des outils utiles aidant au diagnostic. C'est pourquoi, de nos jours, on constate une demande croissante de logiciels de traitement et de visualisation d'images.

La segmentation des lésions d'IRM cérébrale est nécessaire au diagnostic de la sclérose en plaques. En outre, la charge lésionelle est également utilisée dans le suivi des patients atteints de la sclérose en plaques. Dans notre méthode, la segmentation et la caractérisation des tissus sains du cerveau dans des IRM multi-spectrale est la première étape en vue de les différencier des lésions. Toutefois, les méthodes automatiques de segmentation d'IRM du cerveau sont des procédés de traitement d'images très lourds et coûteux. Leur déploiement sur une grille de calcul permet de fournir une solution efficace à ces problèmes.

Dans cette étude, une implémentation efficace de la méthode de segmentation des IRM du cerveau, sur la grille de calcul EGEE, est proposé. Ce travail a été réalisé dans le cadre du projet NeuroLOG qui vise d'une part à fédérer des données médicales et des algorithmes, et d'autre part à partager des ressources informatiques sur la grille.

Une description des outils et des étapes nécéssaires à la transformation de ce type d'application d'imagerie médicale en un flux de données est fournie. Gardant à l'esprit que toute connaissance de la méthode est nécessaire afin de ne pas commettre d'erreur de traitement. Pour se familiariser avec cette dernière, de nouvelles fonctionnalités ont été implémentés dans le logiciel SepINRIA. Ce logiciel est dédié à la sclérose en plaques et est fondé sur les mêmes algorithmes.

Par ailleurs, dans cette étude, nous soulignons la possibilité et l'intérêt du parallélisme, qui est permis grâce à la structure des flux de données. De plus, comme différents types de parallélisme sont possibles, il est important de définir des stratégies d'itérations afin de structurer le flux de données. Ces stratégies sont réalisées grâce à des opérateurs qui relient les entrées et les sorties des différents services qui composent le flux de données.

Les résultats obtenus à l'aide de la grille ont été validés en les comparant aux résultats obtenus en utilisant le "scripte" local sur un seul ordinateur. Puis, les performances en terme de gain de temps et une description des services mis à la disposition de la communauté NeuroLOG sont présentées. Enfin, gâce à la puissance de la grille, des tests sur un parametre d'une méthode de segmentation ont été réalisés, afin d'évaluer son influence sur les segmentations résultantes. L'objectif étant de trouver le meilleur compromis entre vitesse de calcul et exactitude des résultats. Enfin, ce travail montre également les limites de la grille dues à sa structure.

Contents

1 Introduction **1**
- 1.1 INRIA and the Asclepios team . . . 1
- 1.2 SepINRIA . . . 3
- 1.3 The Neurolog ANR project . . . 3
- 1.4 Conclusion . . . 4

2 Medical context **5**
- 2.1 Brain Atlas . . . 5
- 2.2 Multiple Sclerosis disease . . . 6
- 2.3 MRI and Multiple Sclerosis . . . 7

3 Multiple Sclerosis Brain MRI segmentation **11**
- 3.1 Brain MRI segmentation pipeline . . . 11
  - 3.1.1 Spatial normalization . . . 11
  - 3.1.2 Atlas registration . . . 12
  - 3.1.3 Skull-stripping . . . 12
  - 3.1.4 Intensity normalization . . . 13
  - 3.1.5 Brain classification . . . 14
- 3.2 Applications . . . 15
  - 3.2.1 Lesions segmentation . . . 15
  - 3.2.2 Atrophy measurement . . . 16

4 SepINRIA: Multiple sclerosis brain MRI visualization, comparison and analyze Software **17**
- 4.1 Presentation of the software . . . 17
- 4.2 Structure based on C++ . . . 18
  - 4.2.1 Software dependencies . . . 18
  - 4.2.2 Functionality structure . . . 19
- 4.3 My contributions . . . 19
  - 4.3.1 Brain atrophy evaluation functionality (automatic mode) . . . 20
  - 4.3.2 Image fusion and DICOM export . . . 23

5 Brain MRI segmentation pipeline deployment on the EGEE-Grid **25**
- 5.1 Introduction to The EGEE grid and the used tools . . . 25
  - 5.1.1 The EGEE grid . . . 25
  - 5.1.2 Web-Service and Service-Oriented Architecture . . . 26
  - 5.1.3 Workflows . . . 28
  - 5.1.4 Softwares: MOTEUR and Taverna . . . 30
- 5.2 From the pipeline to the workflow . . . 30
  - 5.2.1 Splitting the pipeline . . . 31

5.2.2 Web-Service generation . . . 31
5.2.3 Workflow structure and iteration strategies creation . . . 33
5.3 Results . . . 35
5.3.1 Results validation . . . 35
5.3.2 Shared services . . . 35
5.3.3 Time performances . . . 37
5.4 Medical image application . . . 38

6 Conclusion **43**

A Figures **45**

Bibliography **51**

Chapter 1

Introduction

Contents

1.1 INRIA and the Asclepios team 1
1.2 SepINRIA 3
1.3 The Neurolog ANR project 3
1.4 Conclusion 4

Progress in medical image acquisition systems leads to new possibilities. Different types of images can be acquired and need different types of processing. They are useful in a large range, going from image interpretation and diagnostics to computer-aided and robotic major surgery. All these applications require images processing. Thus, medical image processing domain is also growing to provide clinicians with competitive tools.

In this context, my end of study intership in the Asclepios team, from the INRIA Sophia-Antipolis, was in the context of the Multiple Sclerosis disease. The aim was to understand the pipeline of Multiple Sclerosis Brain MRI segmentation, which has been developed by the Asclepios team. To understand the all-over pipeline and the different algorithms, I have implemented new functionality in the software SepINRIA which is developed exclusively by the Asclepios team and dedicated to Multiple Sclerosis brain MRI analysis. Then, with the help of the I3S team of the CNRS, based on the University of Nice - Sophia Antipolis, the objective was to deploy this application on the EGEE computational grid in the framework of the NeuroLOG project.

1.1 INRIA and the Asclepios team

INRIA *official description from http://www.inria.fr*

INRIA, the French national institute for research in computer science and control, operating under the dual authority of the Ministry of Research and the Ministry of Industry, is dedicated to fundamental and applied research in information and communication science and technology (ICST). The Institute also plays a major role in technology transfer by fostering training through research, diffusion of scientific and technical information, development, as well as providing expert advice and participating in international programs.

By playing a leading role in the scientific community in the field and being in close contact with industry, INRIA is a major participant in the development of ICST in France. Throughout its eight research centres in Rocquencourt, Rennes, Sophia Antipolis, Grenoble, Nancy, Bordeaux, Lille and Saclay, INRIA has a workforce of 3 800, 2 800 of whom are scientists from INRIA and INRIA's partner organizations such as CNRS (the French National Center for Scientific Research), universities and leading engineering schools. They work in 150 joint research project-teams. Many INRIA researchers are also professors and approximately 1 000 doctoral students work on theses as part of INRIA research project-teams.

INRIA develops many partnerships with industry and fosters technology transfer and company foundation in the field of ICST - some ninety companies have been founded with the support of INRIA-Transfert, a subsidiary of INRIA, specialized in guiding, evaluating, qualifying, and financing innovative high-tech IT start-up companies. INRIA is involved in standardization committees such as the IETF, ISO and the W3C of which INRIA was the European host from 1995 to 2002.

INRIA's major goal for 2008-2012 is to achieve scientific and technological breakthroughs in seven priority domains:

- Modelling, simulation and optimization of complex dynamic systems
- Programming: security and reliability of computing systems
- Communication, information, and ubiquitous computing
- Interaction with real and virtual worlds
- Computational engineering
- Computational sciences
- Computational medicine

The Asclepios team *http://www-sop.inria.fr/asclepios/*

In this context, the Asclepios team belongs to the computational medicine domain. it has three main objectives:

1. Analysis of biomedical images with advanced geometrical, statistical, physical and functional models.
2. Simulation of physiological systems with computational models built from biomedical images and other signals.
3. Application of previous tools to Medicine and Biology to assist prevention, diagnosis and therapy.

To achieve these objectives, the research work is divided into four themes:

- Medical Image Analysis which regroup segmentation of multimodal magnetic resonance images for: the Cardiovascular system; multiple sclerosis, the lower.
- Biomedical image analysis which regroup study of Meristem growth, imaging study of the ovarian function and robust Mosaicing for in Vivo and in Situ fibered confocal microscopy.
- Computational anatomy which regroup: Bo-invariant means in lie groups; fast and simple tensor processing; statistical shape models; statistical study of the cardiac and brain diffusion tensor images.
- Computational physiology which regroup tumor growth, extrapolating tumor invasion margins for heart radiotherapy and electromechanical model.

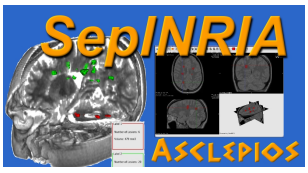
Studied images are anatomical and functional images from: conventional radiology imagery, X-Ray Computed Tomography, Magnetic Resonance Imaging (anatomical, functional and angiographic MRI), Isotopic Imaging (Spect and PET), etc.

To reach these goals, the team can count on several collaborations, either other research team in the world or industrial like Philips Medical Systems, Siemens Corporate Research, etc.

1.2 SepINRIA

SepINRIA[1] is a free software dedicated to Multiple Sclerosis (MS) patient brain MRI visualization, comparison and analysis. It aims at providing to clinicians tools allowing to analyze MS brain images. To be more precise, SepINRIA allows to segment MS lesions (manually or automatically) and to evaluate atrophy (manually or automatically). Images can also be registered together and compared.

This software is available for Microsoft windows XP, Linux Fedora Core and MacOSX. SepINRIA is based on ITK, VTK, wxWidgets, vtkINRIA3D libraries and the MedINRIA framework.



1.3 The Neurolog ANR project

NeuroLOG[2] is a three years scientific project (2007-2009) funded by the french ANR (National Agency for Research) under contract number: ANR-06-TLOG-024. This project designs an

[1] SepINRIA Web-site, http://www-sop.inria.fr/asclepios/software/SepINRIA/
[2] NeuroLOG Web-site, http://neurolog.polytech.unice.fr

ambitious neuroscience middleware, gaining from many existing components and learning from past project experiences. It is targeting to federate medical data, metadata and algorithms, and sharing computing resources on grid infrastructure [Montagnat 2008]. It particularly aims to applications belonging to three different pathologies:

1. Multiple Sclerosis: Challenges are early diagnosis and evolution prediction.
2. Brain Stroke: Challenges are lesions volume variation and anatomo-functional relation.
3. Tumours: Challenges are tumours classification, impact of chemotherapy and anatomo-functional relation.

These applications involve common processes that can be shared by the different partners (*C.f.* Table 1.1).

- **Software technologies, grid infrastructure knowledge.**
- **Databases and ontology knowledge.**
- **Medical imaging process and software knowledge.**

Table 1.1 - ***Map displaying the localization and the work area of the different partners of the NeuroLOG project.***

1.4 Conclusion

In this report, I will first explain the MS clinical context and the interest of brain MRI. Afterwards, I will describe the treatment process done on these MRI by the Asclepios team.

Then, I will present the development I realized into the software SepINRIA. The update of the software to this new release led to the writting of an article. Finally, I will develop my work for the NeuroLOG project on the EGEE grid and confront some results. This study has led also to the writting of an article and the sharing of seven Web-services with all the partners of the NeuroLOG project.

Chapter 2

Medical context

Contents

2.1 Brain Atlas 5
2.2 Multiple Sclerosis disease 6
2.3 MRI and Multiple Sclerosis 7

2.1 Brain Atlas

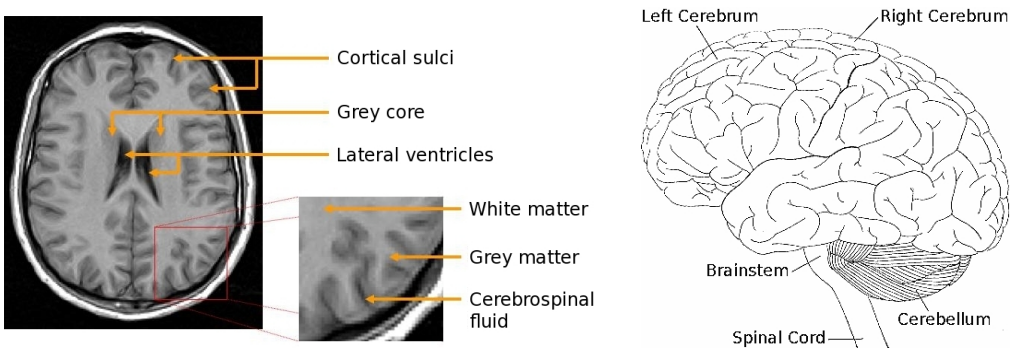
The central nervous system (CNS) contains the majority of the nervous system. It is consisted of the brain, protected by the skull, and the spinal cord, protected by the vertebrae. The CNS is immersed in the cerebrospinal fluid (CSF) which is a solution acting as a buffer for the cortex, providing also a basic mechanical and immunological protection to the brain inside the skull.

Grey matter (GM) and White matter (WM) are components of the CNS. GM consists of nerve cell bodies (neurons) and glial cells. Because of the capillary blood vessels and the neuronal cell bodies, it has a gray brown color. This part of the CNS treats the nervous information in order to create response to the stimulus whereas WM is composed of nerve fiber (axons) covered up by myelinated nerve cell. These cells connect gray matter areas of the brain to each other, carrying on nerve impulses between neurons (*C.f.* Figure 2.2).

The human brain is consisted of three main structures (*C.f.* Figure 2.1):

- The cerebrum is the largest part of the brain and is divided into two hemispheres (left and right). It's surface, named the cerebral cortex, is composed of six thin layers of neurons (gray matter) which sit on top of a large collection of white matter pathways. The cerebrum directs things like perception, imagination, thought, judgment and decision.
- The cerebellum is the part of the CNS regulating the sensory perceptions, the coordination and the motor control. It can be found at the base of the brain and it's composition is similar to the cerebrum.
- The brainstem is the lower part of the brain, creating the link between the cerebral cortex, white matter and the spinal cord. It contributes of the control of breathing, sleeping and circulation.

Other important areas in the brain are the ventricles. Indeed, a number of cavities called ventricles are present in the brain. Ventricles are filled with CSF, which is produced within the ventricle wall.



(a) Brain Components visualization on a IRM (b) Brain atlas representation

FIGURE 2.1 - ***Human brain Atlas: representation of the brain main structures and different components.*** *(Source of Figure 2.1(b): Levinson, medical center for learning disabilities, http://www.dyslexiaonline.com/information/brain/cerebellum.html)*

2.2 Multiple Sclerosis disease

Multiple sclerosis (MS) is a nervous system disease that affect the CNS, leading to demyelination. It's first description has been done in 1868 by Jean Martin Charcot [Charcot 1898]. This disease affect usually young adults from 20 to 40 years old and generally women twice as much as men. Nowadays, it is believed that about 80 000 persons (∼1/1000) are affected in France and 2000 more per year[1].

Symptoms and diagnosis: Multiple sclerosis damages the myelin by thinning the sheath or completely destroy it. When the myelin is lost, the neurons can no longer effectively conduct their electrical signals. The name of multiple sclerosis refers to these scars (scleroses or also known as lesions) in the WM (*C.f.* Figure 2.2). The cause of this disease is still unknown.

Because lesions can appear everywhere in the CNS, the symptoms can completely vary from a subject to another. It can goes from difficulty in moving to problems in speech or weakness and visual problems. That is why the diagnosis of this disease is not easy. The diagnosis of MS can be done using:

- **Clinical data**: measuring the speed of the brain responses using visual evoked potentials.
- **Laboratory data:** testing the CSF can provide evidence of chronic inflammation of the CNS.

[1] Statistics from *http://www.chu-bordeaux.fr/chub/?id=533*

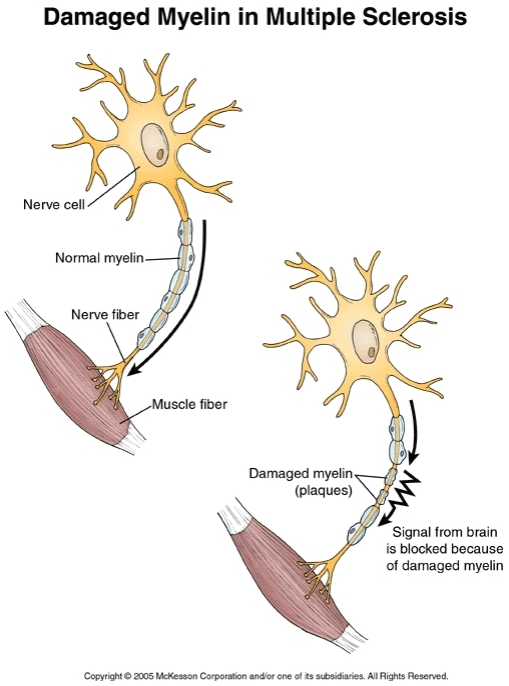


FIGURE 2.2 - ***MS damages the myelin sheath, affecting the conduction of nerve impulses.*** *(Source: University of Michigan Health system website, http://www.med.umich.edu/)*

- **Radiologic data:** using MRI which can show areas of demyelination (lesions) as bright spots on the image. Indeed, visualization and position criterion of lesions in the brain have been establish to determine the presence of MS or not. These criterion have been established by McDonald in 2001 and then fill out by Polman in 2005 [Polman 2005].

Clinical subtypes: Multiple sclerosis is a lifelong illness that can follow different patterns. Either in discrete attacks (relapsing forms) or slowly accumulating (progressive forms). Most subjects are first diagnosed with a relapsing-remitting form which progress throw a secondary-progressive form after 10 years. Between attacks, symptoms can completely disappear, but permanent neurological problems often persist (*C.f.* Figure 2.3).

Treatment: Although this disease does not have a cure, several therapies are helpful. These treatments attempt to return function after an attack, prevent new attacks, and prevent disability. However, medications used have several adverse effects and/or can be rejected from the body patient. Many possible therapies are still under investigation.

2.3 MRI and Multiple Sclerosis

Magnetic resonance imaging (MRI) is a medical imaging technique based on the nuclear magnetic resonance (NMR). This physical phenomenon was described by Felix Bloch and Edward Millis Purcell in 1946. The technique was then refined by Paul Lauterbur in 1970. The

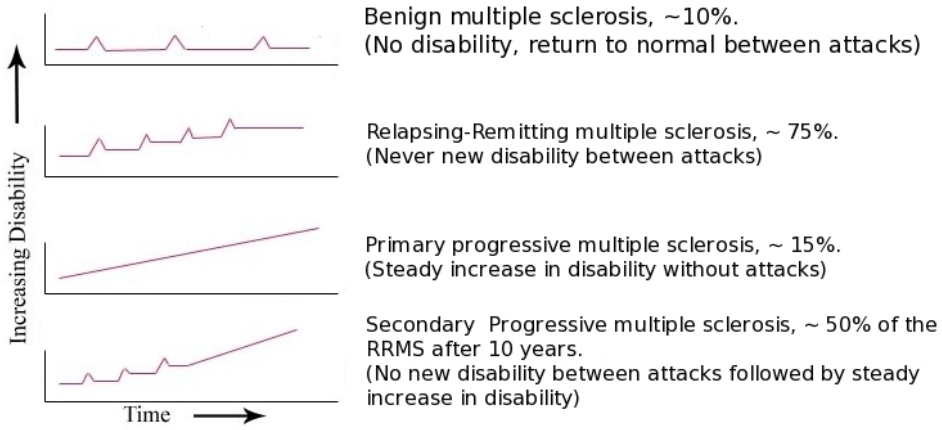


FIGURE 2.3 - ***MS can follow different patterns over the time. These patterns have not the same probability to appear.*** *(Source: The Imaging Informatics Group at the University of Calgary website, http://www.imaginginformatics.ca)*

first MRI on a human was done in 1977.

An MRI can be assimilated as an image in 3D. That is to say a matrix in 3D in which values are assimilated to the intensity. In 3D image, a voxel is the smallest volume element unit (analogous to a pixel in 2D image).

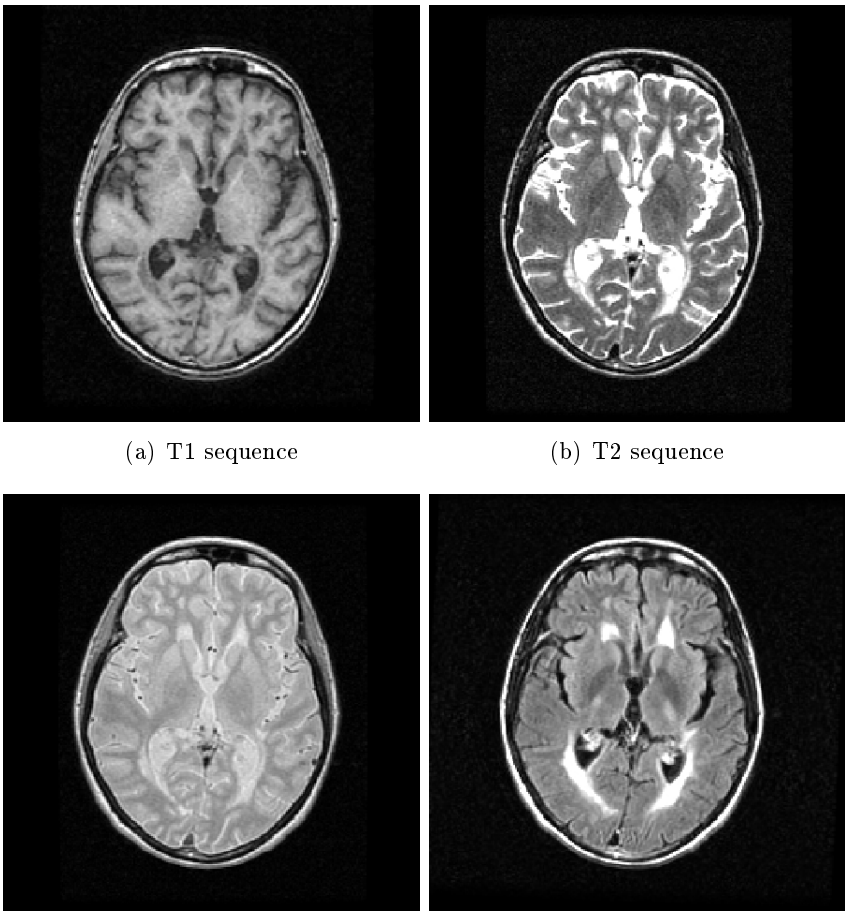
Principle: The Magnetic resonance imaging is divided in four main steps. In MRI, this pipeline is applied to the proton of hydrogen which can be found in water molecule in the human body. Thus, theses proton could be assimilated to small magnets. A whole explanation of the process can be found in [Lauterbur 1973].

In this report, we will only superficially keep in mind that the patient is placed in an electromagnetic field in order to displace the proton from their steady state. Then after the passing of an electromagnetic wave with the frequency of resonance, the proton tend to return to steady state position. This *"relaxation"* generate an other electromagnetic wave which can be measured. Thus, the measure of MRI correspond to the time of relaxation of this signal. This time depend on the intensity of the field and to the nature of the tissue [Liang 1999].

MRI sequences: The time of relaxation of the signal generated by the proton returning to steady state is used among other to generate multi-modal MRI sequences. Indeed, detecting only the proton that have return to steady state after a short time of relaxation allow to identify a similar kind of tissues.

In that way, images obtained can be T1-weighted (T1), T2-weighted (T2), or proton density-weighted (PD) (*C.f.* Figure 2.4). As examples: Fat appears bright (voxels with high intensity) on T1-weighted images and relatively dark (voxels with low intensity) on T2-weighted images contrary to fluids where it is the opposite;

In practice, T1, T2 and PD images provide complementary information, so both are important for characterizing pathology. In MS diagnosis, the sequence T2-FLAIR (*C.f.* Figure 2.5(d)) is also used because lesions appears as high signal intensity.



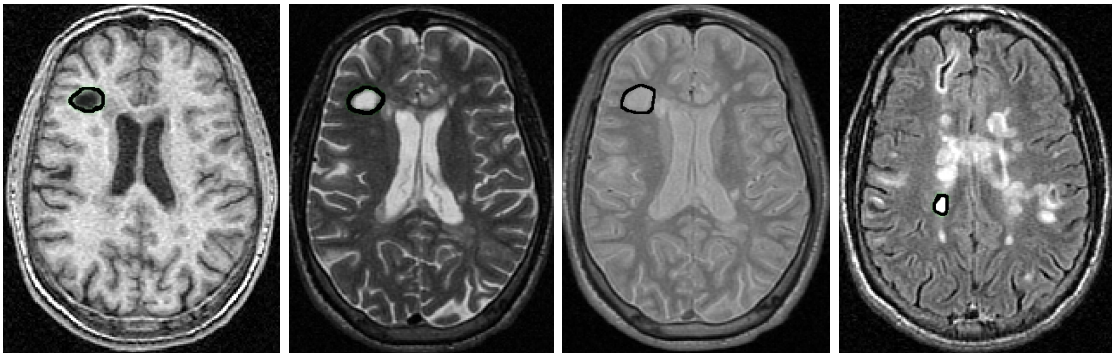
(a) T1 sequence (b) T2 sequence

(c) PD sequence (d) T2-FLAIR sequence

FIGURE 2.4 - ***Examples of MRI sequences from different modalities***

MRI and MS: MRI is superior to other imaging modalities in the imaging of demyelinating diseases such as MS. It is possible to visualize 2-5mm WM lesions and watch their progress over time. They can have different shapes and localizations in the brain. Hopefully, they are particularly visible in T2-FLAIR sequence where they are hyperintense signals (*C.f.* Figure 2.5). However, bony and flow artefacts are also present in the image. That is why, multi-modal MRI sequences are used to isolate these artecfats.

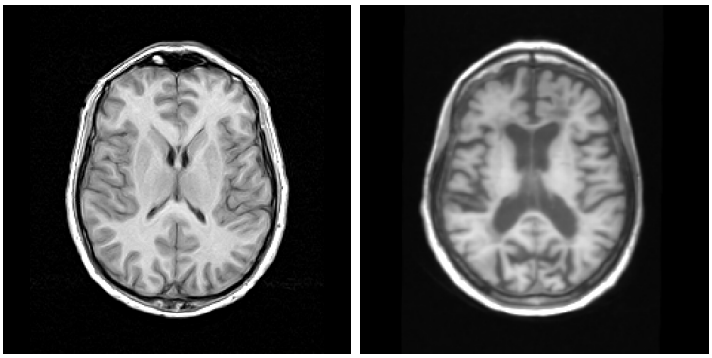
Atrophy: By definition, the atrophy of any tissue means a loss of cells. In brain tissue, atrophy describes a loss of neurons and the connections between them. Thus, we can see in Figure 2.6, that in a subject affected by atrophy, the volume of WM and GM has decrease in favor of



(a) T1 sequence (b) T2 sequence (c) PD sequence (d) T2-FLAIR sequence

FIGURE 2.5 - ***Examples of lesions in MRI sequences.*** *(Source: Images taken from the article [Souplet 2008c])*

the increase of CSF. Indeed if we look more closely, we can see that ventricles and cortical sulci are larger than in a normal patient. Brain atrophy can been seen in different cerebral disease such as MS.



(a) T1 sequence of a normal patient (b) T1 sequence of a subject affected by MS

FIGURE 2.6 - ***Example of brain atrophy effects: Increase of ventricles and cortical sulci volumes.*** *(Source: MRI from the database of the workshop MICCAI'08 MS lesion segmentation Challenge)*

Chapter 3

Multiple Sclerosis Brain MRI segmentation

Contents

3.1 Brain MRI segmentation pipeline 11
3.1.1 Spatial normalization 11
3.1.2 Atlas registration 12
3.1.3 Skull-stripping 12
3.1.4 Intensity normalization 13
3.1.5 Brain classification 14
3.2 Applications 15
3.2.1 Lesions segmentation 15
3.2.2 Atrophy measurement 16

The whole algorithm presented in this chapter has been developed by Dugas *et al* from the Asclepios team, INRIA Sophia-Antipolis, [Dugas-Phocion 2006]. In this report, we make the assumption that we are working on a consistent database of patient images: Each patient can have one or more date of exam and for each instant, MRI sequences T1, T2 and PD are present.

3.1 Brain MRI segmentation pipeline

In this pipeline, the segmentation and the characterization of healthy tissues in multi-spectral MRI is the first step in order to separate them from lesions. This section describes the pipeline used to segment the brain into the different healthy compartments classes (white matter, gray matter, cerebro-spinal fluid).

The input dataset needed for this pipeline is composed of the multi-spectral MRI sequences T1, T2 and PD (*C.f.* Figure 2.4) and images from a reference Atlas (*C.f.* section 3.1.2).

3.1.1 Spatial normalization

As it has been explain in the previous chapter, it is interresting to use simultaneously the different multi-modal MRI sequences. However, this imply to have all these sequences in the same reference frame. Contrary to T2 and PD sequences which are intrinsically co-registered, so in the same reference frame, this is not the case of T1 which has moreover a higher resolution.

The difference of reference frame can be explain by the fact that sequences are not acquired at the same time. Thus, the patient could has move.

To correct this, registration method are used to compute the displacement between two images and register them in the same reference frame. Different kind of registration methods exist. They can either use geometric pattern to find correspondence between the images or they can use the intensity of the voxels [Hill 2001]. In this pipeline, a rigid registration of T1 on T2 sequence is performed using the Baladin algorithm [Ourselin 2000]. Figure 3.1(a) is an example of this step.

3.1.2 Atlas registration

Probability of each voxel to belong to one of the healthy tissue compartments are needed in further steps of the algorithm. The MNI atlas[1] (Montreal Neurological Institute) gives such probabilities.

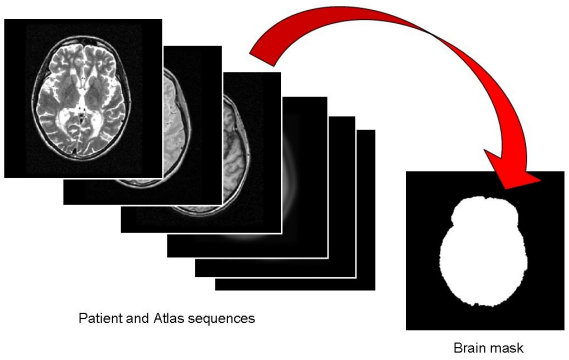
This atlas well match the definition of the noun "atlas" which can be found on Internet: *"a bound collection of maps often including tables, illustrations, or other text"*[2]. As insofar as it provides T1, T2 sequences, illustrating the standard patient and a prior probability of each class. These image have been created on a database of 305 IRM T1 [Evans 1992].

In order to use them, patient and atlas images have to be in the same reference frame. As in the previous part, the registration is perform using the Baladin algorithm but the fact that standard patient images doesn't feet perfectly the images of our patient, generate much complication in the registration. That's why, in this step, a rigid registration followed by an affine one of the Atlas T2 on the T2 of the patient is performed.

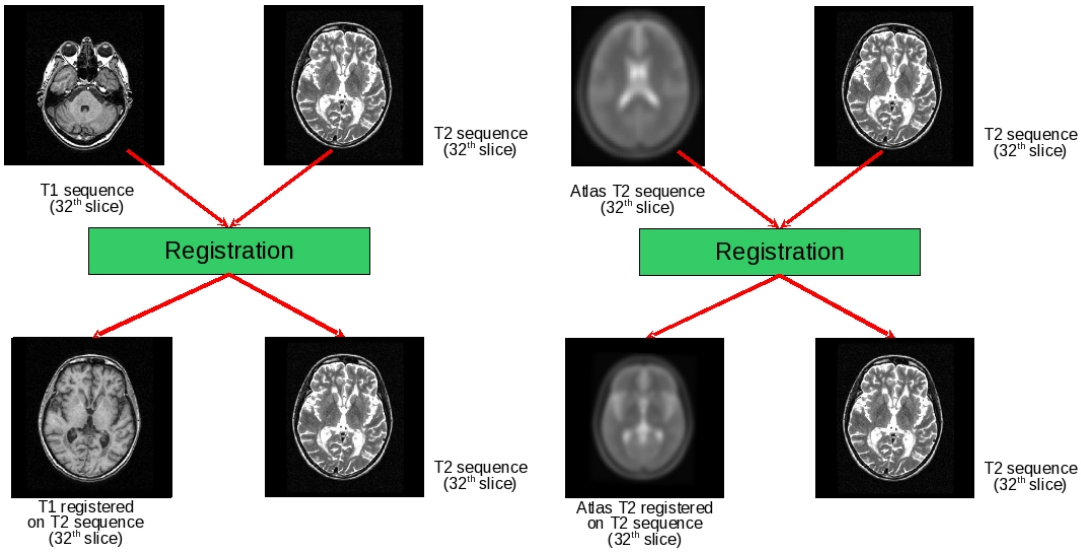
Once the transformation matrix has been generated. It can be applied to all Atlas images. Figure 3.1(b) is an example of this step.

3.1.3 Skull-stripping

At this stage, it is preferable to isolate the brain healthy compartments to the rest of the brain image (tissues, skull, eyes, etc...). Indeed, keeping all the brain can disorder the classification step. Several method of skull-stripping can be found in the literature. Methods' reference and comparison are available in [Souplet 2007].



[1] http://www2.bic.mni.mcgill.ca
[2] Definition from http://wiktionary.org



(a) Registration of T1 on T2 sequence

(b) Registration of Atlas T2 on T2 patient sequence

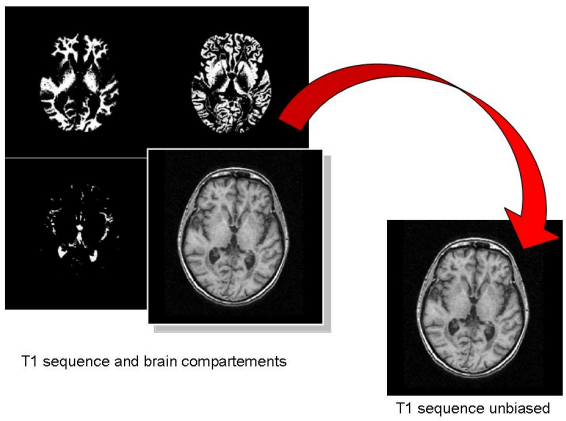
Figure 3.1 - ***Simplify scheme of T1 and Atlas T2 on T2 sequences registration.*** 32^{th} ***slice of each sequence is displayed before and after registration.***

In this pipeline, the skull-stripping is based on a Expectation-Maximization method. T1, T2 and PD sequences as well as the probabilistic brain class images are needed in this step to identify the boundaries of the mask we want to keep.

3.1.4 Intensity normalization

MRI images are often affected by bias [Sled 1998]. Indeed, two voxels belonging to the same brain compartment class can have different intensity.

To correct this bias, a first classification of the brain into WM, GM and CSF classes is realized using the Expectation-Maximization (EM) method and the multi-modal MRI sequences. Then, these segmentations are used to calculate a polynomial per MRI sequence, which will be used to correct the bias [Prima 2001].



3.1.5 Brain classification

Finally, the EM framework is used once again to classify brain MRI voxels from the unbiased sequences. In MRI, the distribution of the voxels intensity can be modelized by a gathering of Gaussian curves. Each brain class will thus be define by a set of parameters: mean and covariance matrix. It is presume that brain tissues can be divided in WM, GM, CSF and PVE classes.

Partial volumes effect doesn't refere to a brain compartment. In fact, voxels can be on the limit between two tissues. Thus, the intensity of these voxels will be a mixture of two intensities. In brain MRI this effect appear, for example, along the limit between the cerebro-spinal fluid and grey matter (*C.f.* Figure 3.2(c)).

To classify each voxel into these classes, the EM algorithm is divided in two steps (a complete explanation of the algorithm can be found in [Dugas-Phocion 2004]):

- **Expectation step:** This step correspond to the labelization of the image. that is to say to calculate the probability of each voxel to belong to each class in function of class parameters and a prior atlas.
- **Maximization step:** This second step consists in the estimation of the Gaussian parameters for each healthy tissue compartment class using the probabilities computed in the Expectation step.

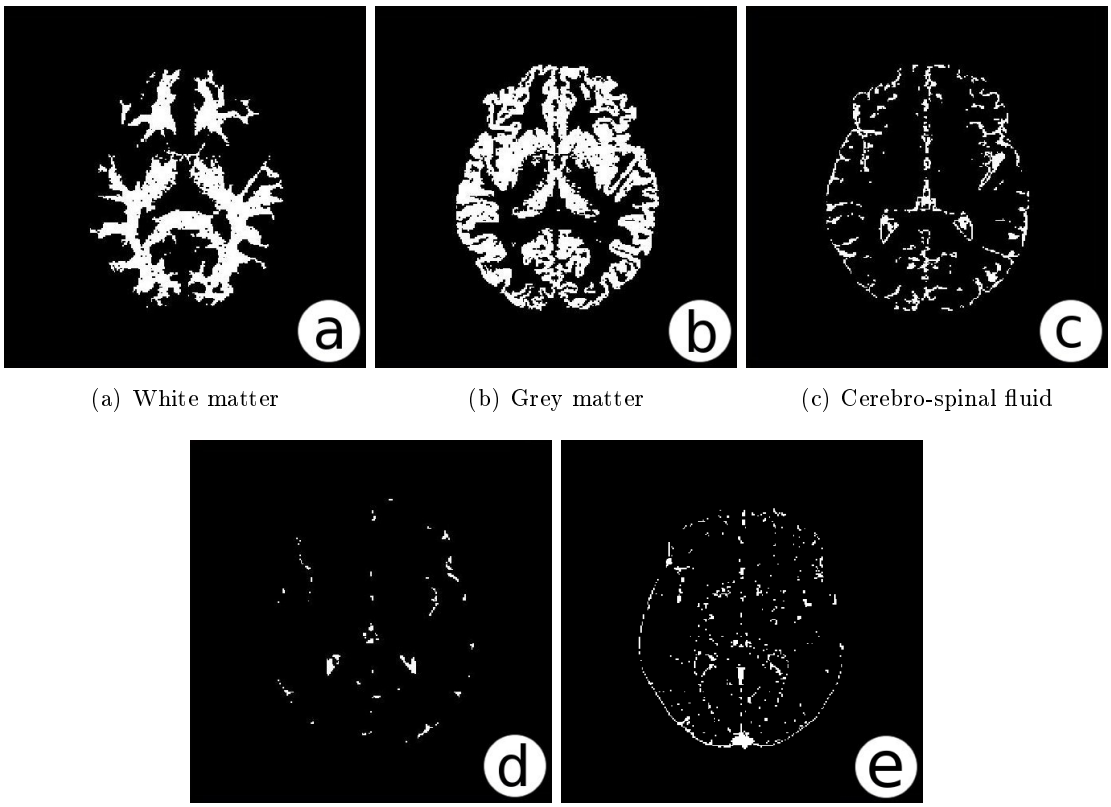
Afterwards, MRI voxels are classified to the most probable class using the computed Gaussian parameters. This lead to the segmentation of WM, GM, CSF and PVE (*C.f.* Figure 3.2). In this Maximization step, outliers could be detected. Indeed, the Mahalanobis distance between the intensity vector (intensity of the voxel in the different sequences), ν, of each voxel and the mean vector of each class k, μ_k (*C.f.* equation 3.1). If the distance is greater than a threshold, the corresponding voxel is labeled as an outlier. Thus, it will not be use in the next step to estimate the class parameters [Dugas-Phocion 2006].

$$d_k = (\nu - \mu_k)^T \Sigma_k^{-1} (\nu - \mu_k) \tag{3.1}$$

Finally, to solve the problem of partial volumes effect and thus obtain the real segmentations of healthy brain compartments. PVE voxels are dispatch between GM and CSF in function of their intensity. Then all segmentations are binarized.

Ratio parameter: In the Expectation-Maximization method, a ratio parameter define the fraction of voxel to be used (*i.e* to be labelized and then providing probabilities for the Maximization step). the relation between ratio value and the percentage of considered voxel is given by equation 3.2.

$$\text{percentage of voxel considered} = 100 * \frac{1}{\text{ratio parameter}} \tag{3.2}$$



(a) White matter (b) Grey matter (c) Cerebro-spinal fluid

(d) Partial volumes (e) Outliers

FIGURE 3.2 - ***Brain binary compartment segmentations. These segmentations have been generated using the workflow on the EGEE grid.***

This parameter can be important because the EM is a computationally intensive tool. That's why working only on a percentage of voxel image could be interresting if, and only if this doesn't affect the results. Section 5.4 provides a parameter sweeping experiment, using the grid, to asses the influence of this ratio parameter on WM segmentation.

3.2 Applications

Brain compartment segmentations are useful information which could be used in further step like the lesions segmentation or the evaluation of brain atrophy.

3.2.1 Lesions segmentation

T2-FLAIR is the most appropriate sequence to visualize lesion (*C.f.* section 2.3. Different methods are available in the literature to segment these lesions. They can either be manual, semi-automatic or completely automatic. A steady of art is available in [Souplet 2008c].

In our pipeline, parameters of brain classes in the T2-FLAIR sequence are identified using

the brain segmentations obtained from the EM method. This allows to border the intensity values which belong to the tissues. Besides, keeping only the voxels which have a intensity value upper than a threshold, allows to isolate the lesions, because they are hyperintense signals in this section. However, artefacts are also segmented. To isolate only lesions, an interest region into the brain is define. This region correspond to the WM we should observed if no lesions were present [Souplet 2008a].

3.2.2 Atrophy measurement

In MS diagnosis, brain atrophy are also often studied. To qualitatively measure the atrophy, the Brain Parenchymal Fractions (BPF) is calculated. This measure depends on the brain compartment volume as it is express in equation 3.3

$$\text{BPF} \quad = \quad 100 * \frac{\text{volume(GM)} + \text{volume(WM)}}{\text{volume(GM)} + \text{volume(WM)} + \text{volume(CSF)}} \tag{3.3}$$

The interest of the BPF value is to follow its evolution in the time. That is why our pipeline is related to the brain segmentation one but take into consideration the presence of several exam dates for a patient [Souplet 2008b]:

1. **Spatial normalization:** All Sequences (*i.e.* from each exam date) are registered in the T2 reference frame of the first date.
2. **Atlas registration:** The MNI atlas is registered in the T2 reference frame of the first date.
3. **Skull-stripping:** Brain mask is computed using the MNI atlas and sequences (registered) of the first date.
4. **Intensity normalization:** For Each exam dates, an EM method is applied using the the corresponding sequences to unbias them. Sequences are also temporally unbias in function of first exam sequences.
5. **Brain classification:** A *"global"* EM is done using all sequences of each date at the same time. Thus, class parameter obtained are used to compute brain segmentations for each exam date.
6. **BPF evaluation:** Using segmentations obtain in the EM method, tissues volumes and then the BPF values are calculated.

Chapter 4

SepINRIA: Multiple sclerosis brain MRI visualization, comparison and analyze Software

Contents

4.1 Presentation of the software 17
4.2 Structure based on C++ 18
4.2.1 Software dependencies 18
4.2.2 Functionality structure 19
4.3 My contributions 19
4.3.1 Brain atrophy evaluation functionality (automatic mode) 20
4.3.2 Image fusion and DICOM export 23

4.1 Presentation of the software

SepINRIA is a free software dedicated to Multiple Sclerosis patient brain MRI visualization, comparison and analysis (*C.f.* Figure A.1). It aims at providing to clinicians tools allowing to analyze MS brain images. SepINRIA website can be found at:

http://www-sop.inria.fr/asclepios/software/SepINRIA/

Full documentation as well as the last release (1.7.2) of the software can be downloaded at this address. The software is currently available on Linux Fedora Core, Mac and Windows XP Operating System.

As we will see in this chapter, SepINRIA is based on C++ language. More precisely on ITK, VTK, wxWidgets and vtkINRIA3D libraries and on MedINRIA framework. MedINRIA[1] is a more general free software developed within the Asclepios research team, dedicated to medical image processing and visualization.

The goal of SepINRIA software is to provide clinicians with a tool allowing to quantify lesion burden and atrophy and also provide other functionality. Efforts have been made to simplify the user interface, while keeping high-level algorithms.

[1] MedINRIA website: `http://www-sop.inria.fr/asclepios/software/MedINRIA/`

Software functionality: SepINRIA has different functionality which can be loaded from a single main window:

- **Lesion Segmentation Edition:** Manual or semi-automatic segmentation of MS lesions (*e.g.* a segmentation realized by an expert). Segmentations can be saved and visualized in 2D or 3D. Lesion number and lesion volume can be computed and print.
- **Automatic Lesion Segmentation:** Automatic segmentation of MS lesions from four MRI sequences (Dual Spin Echo T2-PD, T1, T2-FLAIR).
- **Images or Segmentation Comparison:** Quantitative comparisons of two images registered to assess evolution and comparison between a segmentation (e.g. automatic segmentation) and a segmentation of reference (segmentation of an expert): by computation of the difference image or by visualizing them in the same window (side to side or image fusion).
- **Brain Atrophy Evaluation:** Manual and automatic evaluation of the brain atrophy. Linear measurements computation of the brain, lateral ventricle and third ventricle width are available in the manual mode. And evolution of the BPF in function of the exam dates can be perform in the automatic mode.

4.2 Structure based on C++

This section describes concisely dependencies of the software and also its structure by providing a simplify UML scheme of the code architecture.

4.2.1 Software dependencies

SepINRIA is based on several C++ libraries (*C.f.* Figure 4.2). ITK[2] and MIPS[3] contain both image processing tools. The first one can be downloaded on Internet and is especially used for image conversion. Whereas the second one is inner to the Asclepios team (algorithms presented in chapter 3 can be found in this library).

FIGURE 4.1 - *Used libraries and framework in SepINRIA*

The display is supported by the libraries VTK[4] and vtkINRIA3D[5] (*C.f.* Figure A.1) while the user graphical interface is based on wxWidgets[6]. Finally, the general framework (structure

[2]ITK: http://www.itk.org/
[3]MIPS: http://www-sop.inria.fr/asclepios/software.php
[4]VTK: http://www.vtk.org/
[5]vtkINRIA3D: http://www-sop.inria.fr/asclepios/software.php
[6]wxWidgets: http://www.wxwidgets.org/

and interaction of the functionality in a global frame) has been inherited from the software MedINRIA.

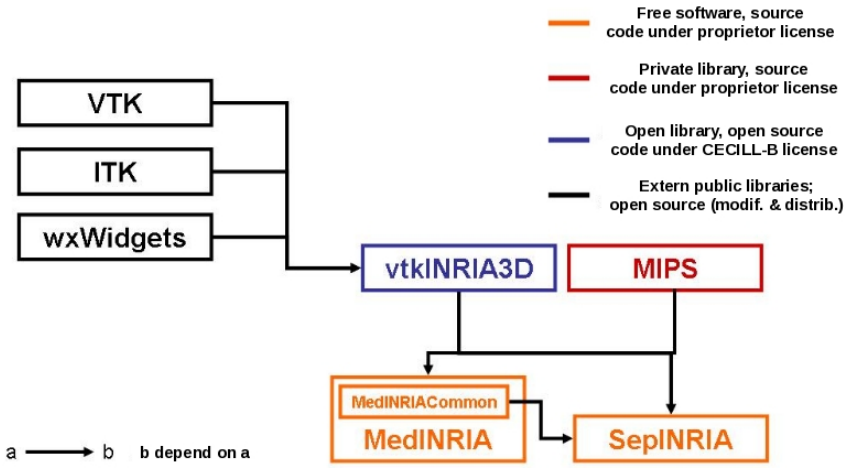


FIGURE 4.2 - *Diagram of SepINRIA dependencies*

4.2.2 Functionality structure

SepINRIA is programmed in the way that each functionality can be independently compiled. That is why, as we can see in Figure 4.3, in the structure, the class SepINRIAapp has got a member of each functionality class named *_module. These last classes allow to compile their own module independently.

SepINRIA uses a database to store images and to write results. All the interaction with the database can be made throw the interface of the software thanks to the database panel available on the left of the SepINRIA application. As this panel should be available in each functionality, its corresponding class "Database panel" is declared independently.

Each functionality of SepINRIA has the same structure. The main frame supervise: the config class which configure the user interface; the process panel, which is a panel interface for data processing and has a member of the database panel class. Then, The class interactor inherit from the main frame and play the roll of interactor between the interface and the factory where processing are done. Finally, this interactor is present in the BAE_module class.

4.3 My contributions

In order to understand the brain MRI segmentation pipeline, I was in charge of implementing the automatic mode of the brain atrophy evaluation functionality. Then, to finalize my job, I also implemented some small tools like the Image fusion and the DICOM export.

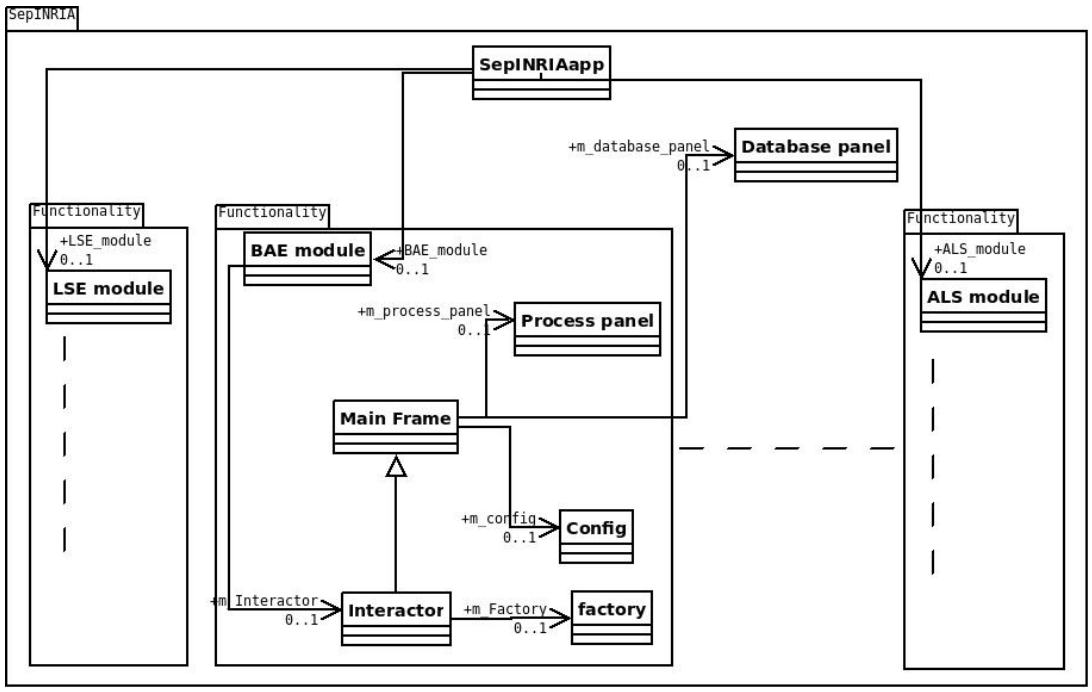


FIGURE 4.3 - *Simplified UML scheme of SepINRIA software structure*

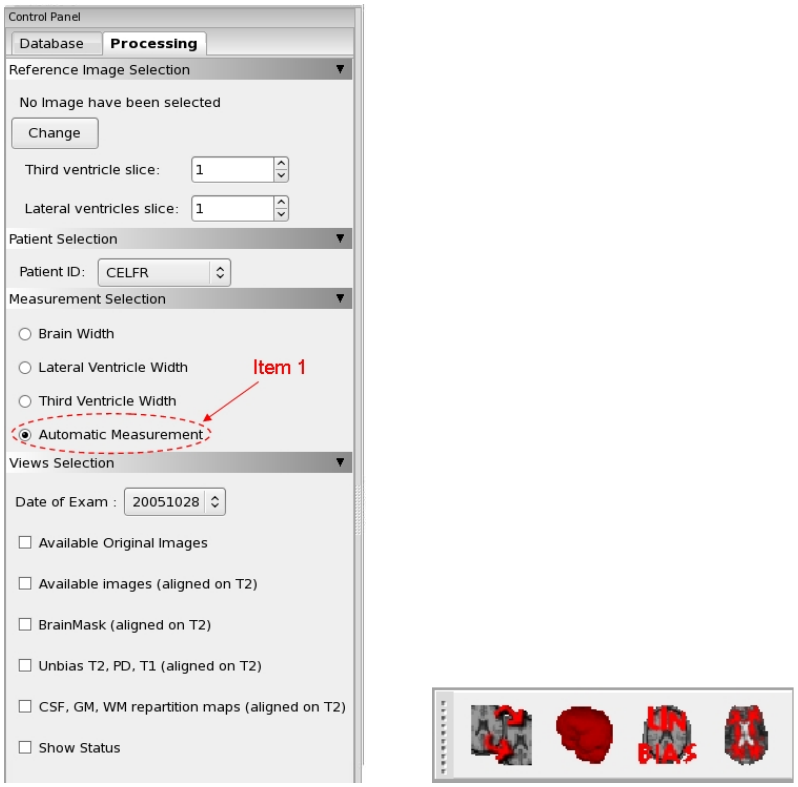
4.3.1 Brain atrophy evaluation functionality (automatic mode)

In this section we won't speak about the manual method and the linear measurements. The aim of this mode is to perform the algorithm saw in section 3.2.2, considering we have a database of patient with different dates containing T1, T2 and DP MRI sequences, and we also have access to the MNI Atlas. The following descriptions derive from the part of the tutorial documentation I have written.

Creation of the interface: As it has been explained, the user interface of a functionality is divided into two parts:

- **Process panel:** The panel belonging to the automatic method is hidden as default value. It can be accessible by selecting the automatic measurement choice (cf. Figure 4.4(a), item 1). This new panel allows to display the images of the different step (original images, registered images on the T2 sequence, the brain mask, unbiased images and the brain compartments) if they have already been computed and finally allows to display a graph of the BPF values in function of the exam dates.

- **Toolbar:** The correspondent toolbar (cf. Figure 4.4(b)) allows to process the different steps of the method until the segmentation of the brain MRI healthy compartment. This can be done step by step or bolt upright.

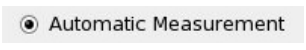


(a) Process panel (b) Toolbar

FIGURE 4.4 - ***Tools (in wxWidgets) belonging to the automatic brain atrophy method. The process panel is apart whereas the toolbar is part of the main frame of the software.***

From the pipeline to the SepINRIA module: The first step to use this method is to select a patient of the database throw the section "Patient Selection" in the Processing panel:

Then to choose the automatic measurement method throw the section "Measurement Selection" in the Processing panel:



At this point, the "views Selection" section will be available in the panel (cf. Figure 4.4(a)) to allow the display of the images of the different step if they have already been done for the selected patient.

- The different steps can be processed throw the toolbar (cf. Figure 4.4(b)) one after another in the right order or bolt upright.

For example, if the step 3 is called, the software check if required step 1 and 2 have been done. And if it is not the case. The software will execute step 1 to 3 in a row.

- After computing one or several step. The correspondent view(s) become(s) available.

- If several exam dates of the patient are available in the Database, it is not necessary to select each exam date in order to compute a step of the method for this date. This is done automatically when executing a step. The choice of the exam date is only used for the display.

The Figure 4.5 shows all the pipeline interactions between the buttons calling processing tools and display.

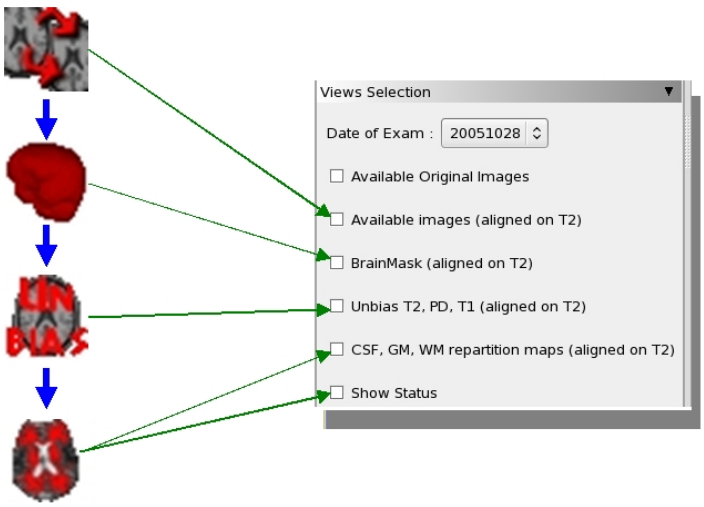


FIGURE 4.5 - *Process scheme of the automatic measurement method illustrated by the button icons of the functionality*

4.3.1.1 Method's steps description

The different step of the algorithm of atrophy measurement saw in section 3.2.2) can be identified behind the interface:

Step 1: Spatial normalization of every sequences in each exam date.

Step 2 & 3: Atlas registration and skull-stripping.

Step 4: Intensity normalization of every sequences in each exam date. And temporel unbias.

Step 5: Brain classification

4.3.1.2 Results

If the workflow has been done correctly on the selected patient, all the results of each step could be display as well as a graph of the BPF values in function of the exam dates (*C.f.* Figure 4.6). This graph can be displayed throw the "show status" option (step 6 of the algorithm of section 3.2.2):

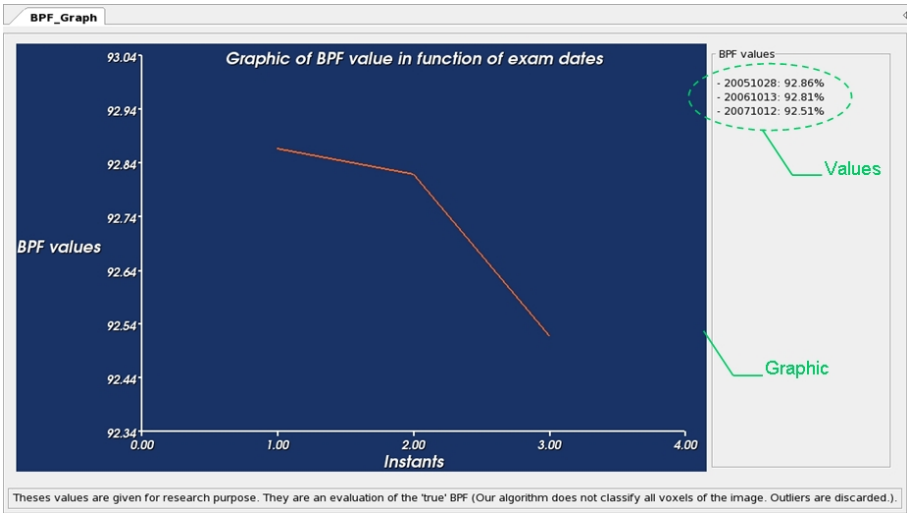


Figure 4.6 - ***Page displayed by the option "show status" of the automatic measurement section, Graph as well as the exact BPF value are displayed. Graph frame automatically adapt itself in function of the number of exam date available for the selected patient.***

4.3.2 Image fusion and DICOM export

Image fusion In the functionality "Images or Segmentation Comparison", after registration of two images/segmentations in the same reference frame, it can be useful to watch them in the

same window in order to visually compare them. That is the reason why I was in charge to develop a display of these images by image fusion. *I.e* Images are set in the same window but with different opacity. Then, playing with the level of opacity allow to partially display either one or the other image (*C.f.* Figure 4.7).

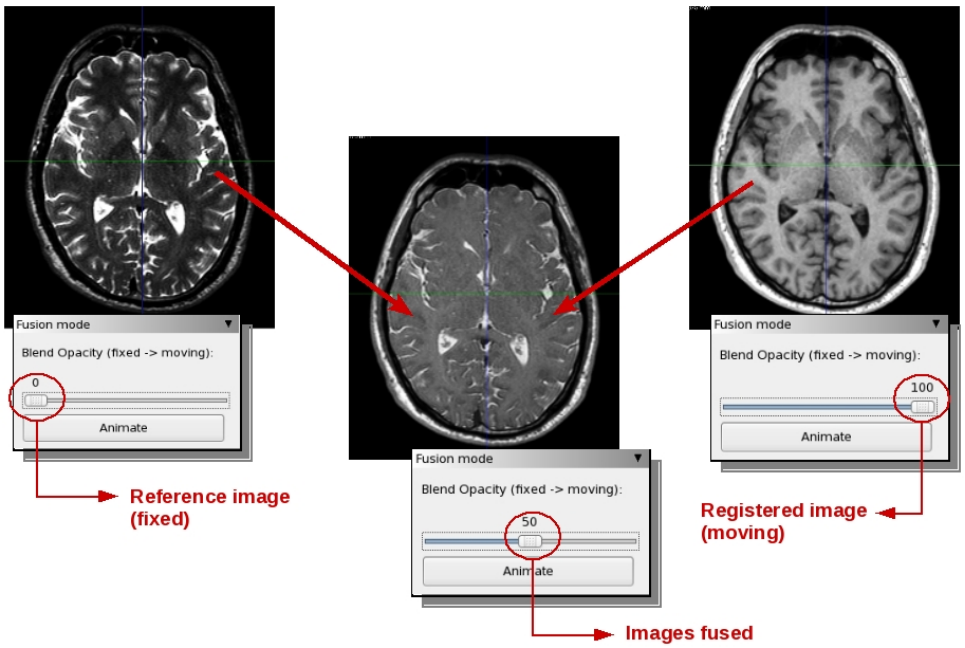


FIGURE 4.7 - ***Comparison of two sequences by image fusion done by the software SepINRIA***

DICOM export Digital Imaging and Communications in Medicine (DICOM) is a standard storage format of data in medical imaging. Indeed, DICOM format containing image as well as medical information can be generated by scanners and then read by standard computer. That is why, this format has been adopted by hospitals.

From our point of view. It is thus important to provide image processing tools supporting DICOM. Therefore, I was asked to implement a DICOM export tool (already available in MedINRIA) in the database panel of SepINRIA. The import tool being already available. This exporter simply allow to choose an image of the database, either new (if just imported) or existing in the database, and save it in DICOM format.

Chapter 5

Brain MRI segmentation pipeline deployment on the EGEE-Grid

Contents

5.1 Introduction to The EGEE grid and the used tools 25
5.1.1 The EGEE grid 25
5.1.2 Web-Service and Service-Oriented Architecture 26
5.1.3 Workflows 28
5.1.4 Softwares: MOTEUR and Taverna 30
5.2 From the pipeline to the workflow 30
5.2.1 Splitting the pipeline 31
5.2.2 Web-Service generation 31
5.2.3 Workflow structure and iteration strategies creation 33
5.3 Results 35
5.3.1 Results validation 35
5.3.2 Shared services 35
5.3.3 Time performances 37
5.4 Medical image application 38

5.1 Introduction to The EGEE grid and the used tools

5.1.1 The EGEE grid

Definition: A computing grid is a network of shared computing and storage resources connected in a grid topology [Foster 2001]. All the work of this report has been done on the Enabling Grids for E-sciencE (EGEE) production grid[1]. EGEE is one of the largest multi-disciplinary grid infrastructure in the world, which brings together more than 120 organizations, acting in different domains ranging from biomedicine to fusion science, to provide computing resource available to the European global research community.

[1] EGEE, http://www.eu-egee.org

A grid infrastructure is ideal for any scientific research, especially for projects where time and resources needed for running the applications are too significant for normal infrastructures.

The EGEE project at present:

$>$ 120 partners
$\sim$ 250 sites in 48 countries
$>$ 68 000 CPU
$>$ 8 000 registered users
$>$ 150 000 jobs per day

Grid computing versus Cluster computing: Even if grid and cluster infrastructure have the same goals, *i.e.* to provide computational and storage resources on a shared network and allow parallelism for applications, differences have to be done:

- **On the structure itself:** Clusters are homogeneous and located on a geographical site while grids connect collections of heterogeneous computers named Computational Element (CE) which are geographically dispersed. CE from a grid can run on different operating system and have different hardware.
- **On the works:** In a cluster, computer works together closely so that they are dedicated to work as a single unit and nothing else. Whereas grids are optimized for applications which consist of many independent jobs which do not have to shared data during the computation process.

Because of these differences, both infrastructure should not be confused. They both have pros and cons: The homogeneity and the geographical proximity of a cluster allow a better control of the network and a fast connectivity between the nodes. But cluster can only be shared by a small community. Whereas, the geographical dispersion of the grid, allow individual users to access computers, databases and experimental facilities simply and transparently, without having to consider where those facilities are located. But this inhomogeneity of the infrastructure can be dangerous in the way that security policies can differ between CE of the grid, and jobs can failed due to the difference of OS.

5.1.2 Web-Service and Service-Oriented Architecture

Using a grid imply to be able to shared every blocks of an algorithm between several computational elements and further more to be able to use the parallelism provided by this infrastructure.

Component: Contrary to object-oriented language like C++ which is highly depending on the operating system and could have internal code dependencies, in component programming,

application are composed at run-time from components selected. And because components communicate only throw an interface, a single component can easily be modified without impacting on the others. The concept of component programming has been introduced by McIlroy several years ago in 1968 [McIlroy 1969]. The main ideas were that software components have to be considered as black boxes, offering families of routine for any given job and these components should be compiled on various architectures without performance loss.

Thus, the term component corresponds to a form of compiled code which can be access by name, reusable and assembled to create an application. Different kind of component and implementation have been proposed: COM/DCOM from Microsoft, Java Beans, etc. These components can be used to implement services.

Service: A service is defined as an autonomous functionality. It is more a concept then a unit of code in the way that the service definition is deployed with the components that composed it. The type of language on which is based the component is independent from the service description. Further more, the same definition can then be used for different implementations. This concept of service is governed by three properties:

1. The interface to the service is platform independent and well defined.
2. The service can be dynamically located and invoked.
3. A service does not call another service.

The two first point allow the distribution of the application without problem of dependencies, whereas the last one ensure no dependencies between different block of the algorithm. Services belongs to Service-Oriented Architecture (SOA).

Service-Oriented Architecture: Finally we come to an infrastructure which could fulfill our requirements. Indeed, SOA describes an informatics infrastructure which allows different applications to exchange data with one another. These functions are divided into basic functionality (services), which are deploy and accessible on a network. Services can be combined and reused and they communicate each other by exchanging data. SOA is basically composed of three actors: The *service provider* runs the service on a particular endpoint (*i.e* a port and Internet address) and publishes its interface (description) in a *service broker*, which allows the *consumer* to discover the service and to invoke it.

Web-service: Services take all their utilities when they are accessible over standard Internet protocols access like HTTP that are independent from platforms and programming languages. This is the goal of Web-services which are the most common implementation of services. The description of a Web-service is done by a Web-Service Description Language (WSDL) file, which is based on XML language and has been standardized by the W3C[2].

[2] The World Wide Web Consortium, http://www.w3.org

A WSDL file describes a Web-Service by specifying two kinds of XML tags. Here are seven of the main XML tags:

Tags describing what has to be invoked:

- Types: types of the exchanged data.
- Message: set of parts (Input/Output parameters).
- Operation: set of messages (equivalent of a method).
- Port type: set of operations (equivalent of a class).

Tags describing how to invoke it:

- Binding: protocol used to invoke the service (e.g. SOAP/HTTP).
- Port: internet address and port.
- Service: set of ports.

In our framework, a generic Web-service description is used. Then, services are wrapped using a Generic Application Service Wrapper (GASW) [Glatard 2006a]. Figure 5.2 is an example of a Web-service description based on our generic format. It creation will be developed in the section 5.2.2.

Then, in our case, Simple Object Access Protocol (SOAP) is used to invoke the Web-services. This protocol allows to call a procedure, physically based on an other computer, to execution on a shared network.

5.1.3 Workflows

Definition: As SOA is a concept, it doesn't describe how to composed the service. Among different composition approaches, workflows have been chosen. In a more generic view, a workflow can be assimilate as a reliably repeatable pattern of activities obtained from a organization of resources. Domains where workflows are used are broad. Thus, the workflow management coalition[3]. proposes the following definition of workflow management:

" Workflow management is the automation of business procedures or "workflows" during which documents, information or tasks are passed from one participant to another in a way is governed by rules and procedures."

In the SOA context, a workflow is a particular type of software composition system, where the participants of the workflow are components to be composed. As we saw in previous section, a component is a small independent application equivalent as a black box. Thus, for the workflow only inputs and outputs have to be known. In this way, creating the workflow consists

[3]The Workflow Management Coalition, http://www.wfmc.org

in connecting input ports of a black box to output of others, thus building an application as a data flow graph.

Parallelism: The aspect of independence is attractive in grid computing because it could be assimilate as parallelism. In fact, three kind of parallelism are available through workflow.

1. Workflow parallelism which corresponds to the concurent execution of two independent data by two independent components.
2. Data parallelism which corresponds to the competitor execution of independent data by a single component.
3. Service parallelism (also denoted pipelining) which corresponds to the concurrent execution of two independent data items by two components linked by a precedence constraint. That is to say, if two linked components C1 and C2 are executed on two data items d1 and d2. When C1 has finished with d1, C2 can start on d1 while C1 can also start on d2.

These case will be illustrated in section 5.2.3:

Classification: Different kind of workflow can be distinguished. Several workflow classifications have been proposed in the literature [Yu 2005]. These descriptions are based on criterion like the workflow structure, composition system or specification. Since our work has been done in collaboration with the I3S team from the University of Nice, we will use the classification done by Tristan Glatard during his thesis [Glatard 2007]. This classification of workflow is based on the presence or absence of functions, data and resources in the workflow representation. It has been realized to help user to choose the workflow definition with the appropriate level of information. This is interresting in the case of medical image application where the user can either be:

- The medical image analysis scientist who develops the workflow and its components.
- The clinician end-user who instantiates the workflow on the data.
- The grid expert who performs the grid deployment and in particular the scheduling of the workflow on the resources.

Five main workflow classes have been distinguished, we provide a quick abstract:

- **Formal workflow:** No information is given about the nature of the implied activities, the amount and type of data processed and the used resources. Suitable for workflow analysis because they offer an abstract representation of the application.
- **Functional workflow:** Only participants and their dependencies are defined. The amount and type of data are only specified at runtime. It remains equivalent as drawing the pipeline of an application.

- **Service workflow:** Both functions and resources are specified. As in functional workflows, the data is not defined and is specified at runtime. Reference to Web-services are done to define resources. This is the class of workflow used in this work.

- **Tasks-graphs:** Both functions and data are defined and mixed. Tasks (function with all the parameters) to be executed are completely defined: the workflow representation specifies their number as well as their nature. This leads to a static representation of the workflow (i.e the number of task is known prior to the execution).

- **Executable workflows:** This case correspond to a tasks-graph workflow which can be scheduled onto resources. For example, time when temporary file produced by the workflow can be deleted can be determine in order to reduce overload of storage unit.

5.1.4 Softwares: MOTEUR and Taverna

In order to describing and structuring the service workflow, Scufl data-flow oriented language is used (derived from XML). This language can be generated using the graphical interface of Taverna[4] (designer [Oinn 2004]).

This Scufl description is divided into two files. First, inputs are specified in an independent *"input"* Scufl file where XML tags indicate the real name and localization of data files.
This file is independent from the workflow description file but the same tag for the inputs have to be use to link the two files. The use of two separate files allow to manage the input datasets without having to change the description of the workflow.

In Scufl, participants of the workflow (component) are called processors. Theses processors have input and output ports that can contain several data items and are connected to other ones with data link. A data link is simply a pipe between an output port of a processor and an input port of another one. As we will see in the iteration of strategies (*C.f.* section 5.2.3), several output ports can be linked to a single input port and similarly, an output port can be connected to several input ports.

Finally, the workflow is efficiently executed on the EGEE grid through the MOTEUR enactment engine [Glatard 2006b] developed by the I3S team. MOTEUR provides a graphical interface hiding to the user the complexity of individual services submissions and management (*C.f.* Figure 5.1). Moreover it has been optimized to first, exploit the parallel resources available on the grid infrastructure. Indeed, the three different kind of parallelism available by workflow are used. And secondly, to group sequential jobs in order to lower the number of services invocations and minimize the grid overhead resulting from jobs submission, scheduling and data transfers.

5.2 From the pipeline to the workflow

This section describes the creation of the MS brain MRI segmentation workflow step by step, from the used pipeline see in section 3.1 to the executable workflow on the EGEE grid (*C.f.*

[4] Taverna, http://taverna.sourceforge.net/

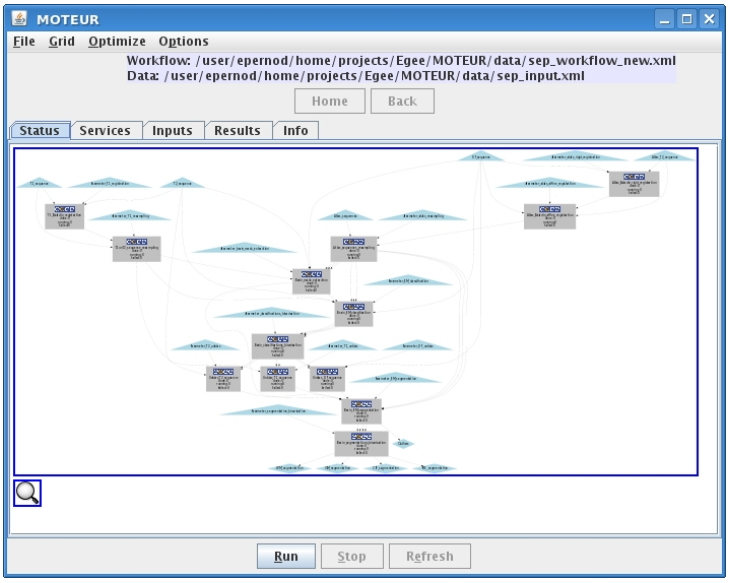


Figure 5.1 - ***View of MOTEUR software interface. As our brain segmentation workflow has been loaded, MOTEUR provides a graphical view of the Scufl file.***

Figure A.4).

5.2.1 Splitting the pipeline

For enabling the execution of service workflow, it is first necessarily to describe our pipeline as a workflow. Concretely, the pipeline has to be splitted into services correctly linked together. Then, iteration strategies have to be described between services in order to structure the workflow.

The first work is to understand the pipeline and then identify the step of the pipeline which can be assimilated as independent one. This work leads to the Figure A.2 in Appendix. We will see that the first approach adopted was to divide the pipeline into different component, reflecting the call of each legacy application. Thus the splitting is more precise than the description of the steps given in 3.1. It leads to the Figure A.3 which looks more like a tasks graph. For each block, inputs and outputs have clearly been defined, in order to define the corresponding service.

As we will see further in this report. It is important to find the right middle in splitting the pipline into services. Indeed, splitting help to parallelize the application and so provide a gain of execution time, but splitting to much could produce loose of time by sending this tasks on the grid.

5.2.2 Web-Service generation

For each service, a Scufl file has been created including: an enumeration of inputs and outputs, as well as the name and the localization of the binary file, corresponding to this service. And if necessarily the name and the localization of the shell file script encapsulating the binary file. The Figure 5.2 shows an example of a generic Web-Service description file and the correspondences

with the application.

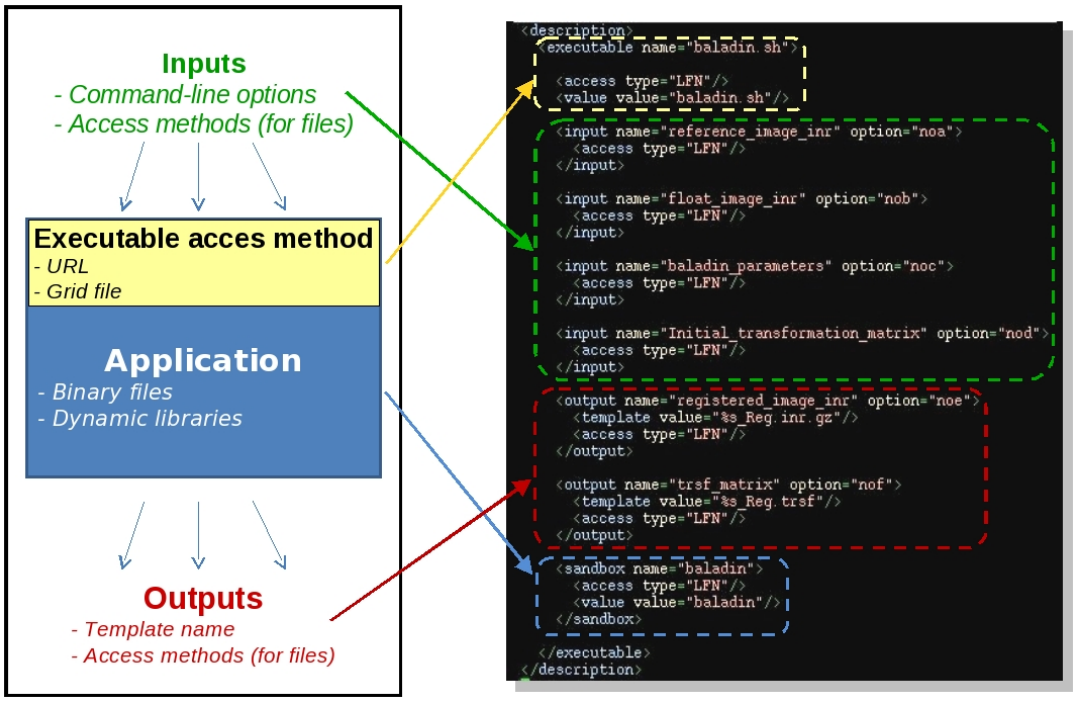


Figure 5.2 - *Scheme of the correspondence between an application and the Web-service description file.*

In practice additional shell scripts are needed in some cases. In fact, binary files may have many outputs or may take fixed filenames as inputs. However, for MOTEUR, outputs from a service have to be listed and are automatically renamed. Consequently, shell scripts provide a good solution to overcome these problems. Furthermore, this is also helpful in case of different calls of the same service's definition. For example, three registrations are needed for the MS lesion segmentation algorithm 3.1 and they don't have the same list of arguments. But only one description of the service is used with a text file (listing the parameters) as an additional input. Then, these files are read in the shell script in order to correctly execute the software. And finally, shell script allow the creation of small pipeline in the case of service based on several binary files.

Before any transformation of the pipeline into a workflow, a first validation of the services has been made. All the services have been independently executed and tested on the grid with a test dataset. Results have then been compared to the ones obtain with the local pipeline.

These tests reveal that even if the software are written in a generic language like C++, compiling it directly on a Computing Element (CE) of the EGEE grid without shared libraries is highly advised to avoid any kind of execution problems (problem cited in section 5.1.1). Indeed, even if we had specified in MOTEUR that only computer with linux OS should be used to run our jobs, there were compatibility problems due to different release of linux kernel between my computer and computer of the grid.

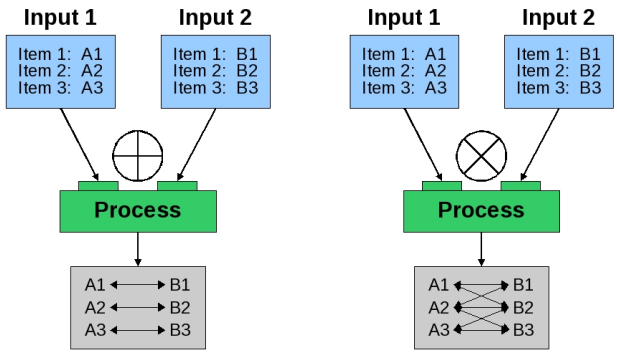
5.2.3 Workflow structure and iteration strategies creation

Then, the software Taverna has been used to structure the workflow:

- Images and parameters text files have been defined as inputs of the workflow.
- The four different healthy compartments brain classes as outputs of the workflow as well as the image of outliers points.

Since Taverna is a graphic interface, a service (called operator in taverna) is displayed as a box with inputs and outputs ports. It is then easy to correctly link services one to each other to reproduce the pipeline if it has been precisely described. For each of these processor, the URL to the Web-Service description file is precised. Several processor can have the same description if the task is the same but at an other step of the pipeline with different data. Figure A.4 (*resp.* Figure A.5) is a scheme of the workflow obtained by Taverna in his simplified (*resp.* complex) version.

Iteration strategies: As we saw in section 5.1.4, outputs of several services can be linked to a single input port of another service. Thus, it is important to define the strategy of composition over the inputs port of a service in order not to mix the data coming from concurrent execution of the workflow. Two different data composition operators have to be considered to generated the iteration strategies. The one-to-one (Dot operator: $\oplus$) and the all-to-all (cross operator:$\otimes$) data composition operator [Montagnat 2006]. Figure 5.3 represents the composition done by the Dot and Cross product (the arrows representing the compositions).

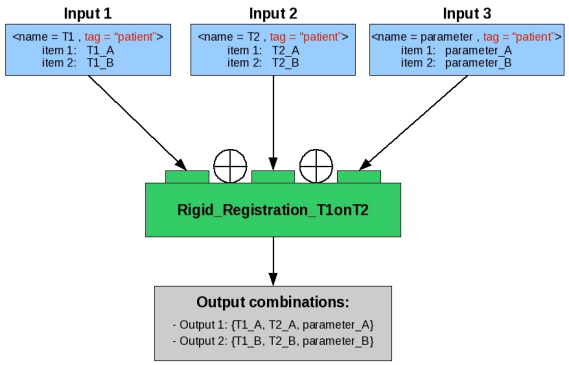


(a) Dot product (b) Cross product

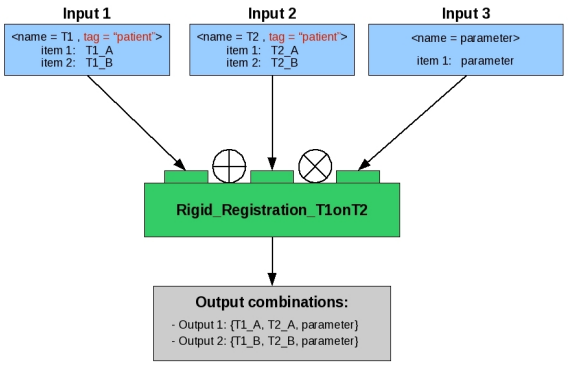
FIGURE 5.3 - ***Example of data compositions with Dot and Cross operators on two input datasets of both three items. Arrow represents an output composition***

In concurrent execution of the workflow, Dot product are used to avoid cross-road composition from images from one patient with images from another. A XML tag is used in the input file to express the fact that tagged inputs are referring to the same patient. Then these inputs are composed with a Cross product with others data of the workflow. For example, in the case of

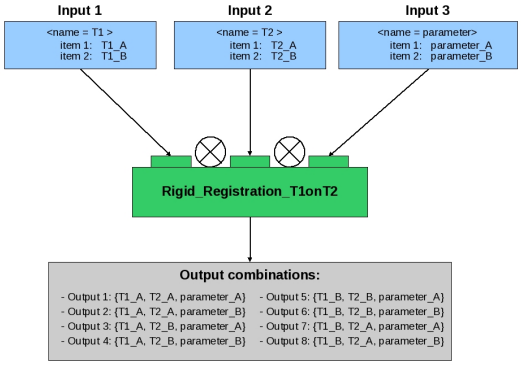
the rigid registration of T1 on T2 sequence. Three different cases are possible with two patients A and B (suffix: _A and _B). They are illustrated in Figure



(a) Only Dot product



(b) Dot and Cross product



(c) Only Cross product

FIGURE 5.4 - ***Example of data composition possibilities with Dot and Cross operators for the rigid registration of the T1 on T2 sequence.***

These configuration have been used to test different value of the ratio parameter of the second EM (*C.f.* section 5.4). Indeed, for the first EM, the parameter file was considered as not belonging to the patient. So it was the case *"Dot and Cross product"*. Whereas, for the second EM, the parameter file was considered as belonging to the patient like in the case *"Only Dot product"*. Acting this way allow, for example, to put two times the same patient but with two different parameter files.

Parallelism use: Good iterating strategy allow to use the parallelism of the workflow without bad supprise. Here are some example of parallelism, saw in section 5.1.3, we used for this workflow (refer to Figure A.4 for the name of the services).

- **Workflow parallelism:** Different sequences T1, T2 and DP can be unbiased in parallel

(execution of *Asclepios_unbias_** services in parallel).

- **Data parallelism:** As soon as the affine registration has generated the transformation matrix between Atlas images and patient sequences. This matrix can be applied to all Atlas images at the same time. (concurrent executions of *(Asclepios_Resampling_Atlas)* on each Atlas image).

- **Service parallelism:** When rigid registration of the Atlas *(Asclepios_Baladin _registration_RigidAtlas)* has been done on the first patient, the affine *(Asclepios_Baladin _registration_AffineAtlas)* one can start on this patient, while a rigid registration on the next patient can also start.

5.3 Results

In this section, we propose: first, a validation of the results obtained from the grid by comparing them to the local pipeline results; then, a final description of the seven services, used to build the workflow, which have been selected to be shared with the other partner of the NeuroLOG project; and finally, a study of time performance of the grid.

5.3.1 Results validation

To compare the results, a ratio parameter equal to 1 has been used (all voxels from the image are taken for the maximization step *C.f.* section 3.1.5). Executions were done with images of 256×256×64 voxels for T2 and PD sequences and 256×256×152 for T1 sequence. For each execution, the four different healthy compartments classes' segmentations (*C.f.* Figure 3.2) have been generated and compared. Comparisons are done by subtracting an image to another. In every case, resulting images were empty, validating the correct deployment of the workflow on the grid.

5.3.2 Shared services

Since the workflow has been validated, it has been decided that seven services being part of the workflow will be shared by the Asclepios team to all the partners of the NeuroLOG project. Thus, partner could use these services in their own application. This imply to provide also a complete description of the services. This section is only an abstract of the complete description that I have written, which contain more specific details. Each service is composed of the Web-service description, the shell script, the binaries and a parameter text file.

Name	**Asclepios_Conversion_service**
Performed Data Processing	Format conversion of Data.
Description	Convert an image from a format to another, without changing the information of the data.

Name	**Asclepios_Baladin_registration_service**
Performed Data Processing	Registration (Rigid or non-rigid affine registration.
Description	Registration of a floating image on a reference one. Rigid and non-rigid (affine) registration are supported. Create a transformation matrix (4x4) and the image in the reference frame of the reference image.

Name	**Asclepios_Resampling_service**
Performed Data Processing	Resampling in an other data grid.
Description	Resample an image according to a transformation based on a Registration dataset (4x4 matrix, or a dense deformation field). Uses either the same sampling grid as input image, or a new one, through the specification of the new image geometry.

Name	**Asclepios_EMBrainMask_service**
Performed Data Processing	Brain tissue segmentation.
Description	Segmentation of brain mask, based on a priori information concerning grey matter (GM), white matter (WM) and cerebrospinal fluid (CSF) coming from an atlas.

Name	**Asclepios_EMBrainClassification_service**
Performed Data Processing	Brain tissues segmentation.
Description	Segmentation of tissues, based on EM classification, and on a prior masks concerning GM WM CSF coming from an atlas. This algorithm takes into account extra classes due to partial volume effect (PVE) between grey matter and CSF.

Name	**Asclepios_Unbias_service**
Performed Data Processing	Estimation of bias field and calculation of debiased image.
Description	Compute Bias field of a sequence. Based on a prior probability information concerning grey matter (GM), white matter (WM). Compute then unbias sequence.

Name	Asclepios_Binarisation_service
Performed Data Processing	Create binary segmentation datasets from probabilistic segmentation datasets.
Description	Create binary segmentation datasets from probabilistic segmentation datasets, based on the highest probability found at each voxel in the 4 images; create one additional segmentation dataset containing all non-classified voxels.

5.3.3 Time performances

A ratio equal to 1 has also been used in the executions used to proceed time performance values. Local executions have been done on a 2.0 GHz computer. We report the mean value over many "one-patient executions". As expected, the local execution time evolves linearly in function of the number of input datasets. But this is not the case for grid execution time (*C.f.* Figure 5.5). Two points discussions can thus be done.

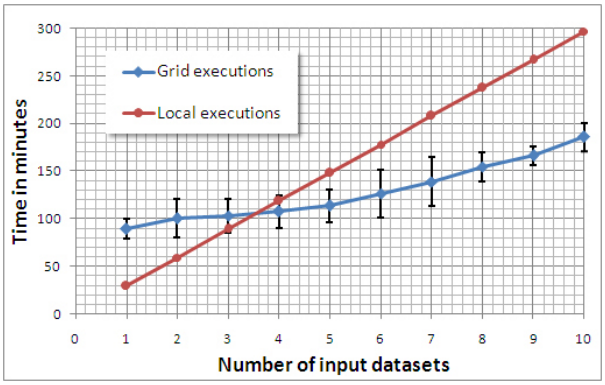


FIGURE 5.5 - ***Mean execution time and its variations with respect of the number of input datasets. Comparison between local execution on a single computer and EGEE grid executions. Concerning EGEE grid executions, all points have been computed 3 times in order to compensate the workload variations.***

- First, the execution time of the workflow can change depending on the resources availability and variations. To address this point, the workflow has been run several time on the same number of input datasets and the mean value has been computed.
- Secondly, the execution time is not constant (as it could be expected in case of complete parallelism). Indeed, the multiplication of inputs generate more transfers, so queue time is more consequent.

Finally, we can observe that for this application, using the grid appear to be efficient for more than 4 or 5 input datasets.

Potential issues: The EGEE grid is actually using the gLite middleware[5] which provides a framework for building grid applications. In this framework, the Resource Broker (RB) is among others in charge of accepting user jobs and then assigning them to the most appropriate CE. This choice is done by selecting CE which, first fulfill the requirements expressed by the user and then have the highest rank.

On the EGEE grid, it appears that the rank is reflecting the response time of a CE. This choice can be discussed because the fastest responding CE is not necessarily the most powerful one, and above all, not necessarily directly available. Hence the workload management becomes sometimes a bottleneck for our application.

Moreover, the workload on the EGEE grid is highly variable thus leading to high and variable latencies and many faults, impacting the total execution time. After a delay, jobs have to be canceled and resubmitted. Optimizing job submission strategies is still on a research stage [Lingrand 2008].

Of course, in between single computer and production grid, we could have envisaged the use of a cluster, improving the execution time for several (5 or 6) datasets without encountering production grid problems. However, this is not in our scope for different reasons. The first reason is we have in focus to be able to support large databases of patients; increasing the number of patients will conduct to cluster saturation. The second reason is that we are targeting final users that do not necessary have access to a cluster, even for small extend experiments; managing a grid access is thus a solution.

5.4 Medical image application

As we saw in section 3.1.5, the Expectation-Maximization method use a ratio parameter. In this section we are targeting to assess the influence of this parameter on the pipeline results (*C.f.* Figure 5.9). In order to clarify the explications we will use the percentage of voxels considered rather than the ratio value in this part (*C.f.* equation 3.2 for the relation).

In fact, by taking only a part of the image voxel, the speed of the algorithm could be improved but it could also affect the accuracy of the resulting segmentations. This is reason for studying the relationship between this percentage and the compromise between accuracy and speed is interesting for further works.

To quantitatively evaluate this impact, WM segmentations have been generated using different percentage of the voxel in the method and have been compared to the segmentation of reference (*i.e.* generated with 100% of the voxels) by computing the sensitivity and the specificity described in equation 5.1.

Sensitivity and specificity: They are statistical measures of the performance of a binary classification test.

[5] Lightweight Middleware for Grid Computing, http://glite.web.cern.ch/glite/

$$\begin{aligned} \text{sensibility} &= \frac{\text{true positives}}{\text{true positives+false negatives}} \\ \text{specificity} &= \frac{\text{true negatives}}{\text{true negatives+false positives}} \end{aligned} \tag{5.1}$$

In our case, given a segmentation of reference (Figure 5.6(a)) and a generated segmentation (Figure 5.6(b)), sensitivity measures the proportion of points segmented which belongs to the segmentation of reference and specificity measures the proportion of points not segmented which don't belongs to the segmentation of reference. According to this, we can propose the following definitions for the different terms, which are illustrated by the Figure 5.7:

(a) Segmentation of reference (b) Segmentation obtained

FIGURE 5.6 - ***Example of two segmentations to compare.***

- **True positive:** Voxel segmented and belonging to the segmentation of reference.
- **True negative:** Voxel not segmented and not belonging to the segmentation of reference.
- **False positive:** Voxel segmented but not belonging to the segmentation of reference.
- **False negative:** Voxel not segmented but belonging to the segmentation of reference.

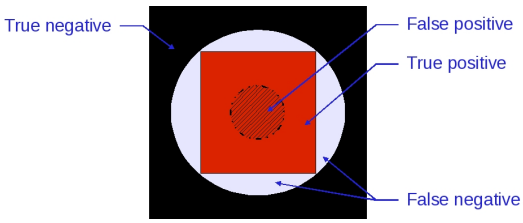


FIGURE 5.7 - ***Example of significations of true/false positives and negatives by comparing Figure 5.6(a) and Figure 5.6(b)***

Executions: These experiments have been done on MRI from one patient affected by Relapsing/Remitting Multiple Sclerosis and one normal subject.

It is important to underline that when taking only a percentage of the voxels, they are chosen randomly in the 3D image. Consequently, different results can be obtained for a same

ratio parameter. To minimize the influence of this effect, many executions have been done with the same percentage of voxels and means values of the sensibility and the specificity have been computed. The Figure 5.8 displays these values in function of the percentage of voxel considered with the variations around mean values.

For this application, the power of the grid provides an efficient help to generate all the results (∼10 executions per percentage of voxel value). Indeed, the ratio parameter was written in an input parameter text file and has been assimilated as relative to the patient (*C.f.* Figure A.4). Acting this way allows us to test all the different ratio parameters on one patient (case "Only Dot product" of the iteration strategies in section 5.2.3) in only one execution of the workflow, by putting as much parameter files as the number of input datasets (*C.f.* Figure 5.4(a)).

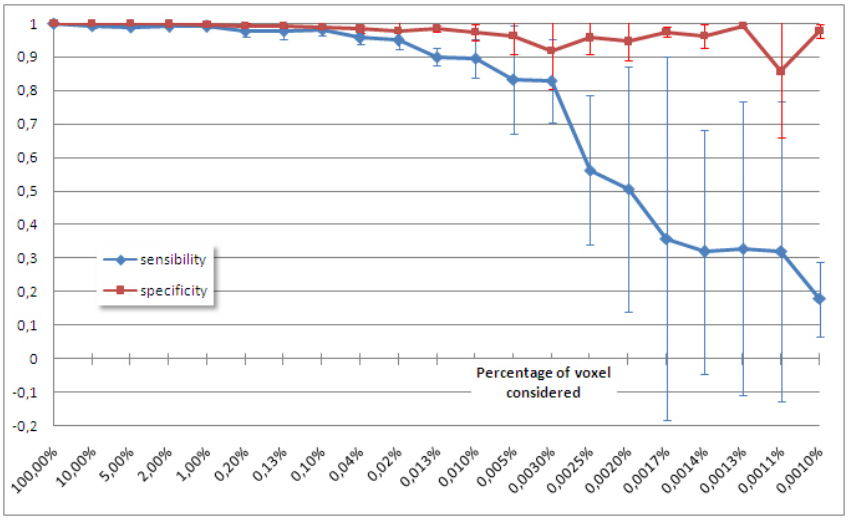
Computing the 210 segmentations should have taken approximately 100 hours on a local computer. whereas, one of the ten execution of 21 input datasets took approximately 4 hours, depending on the overload of the grid. So, it took us approximately 40 hours to compute the 210 segmentations. Thus, we have a gain of time higher than 1/2.

Discussion: Due to the skull-stripping step, the segmentation of the different healthy compartments is done on approximately 830.000 voxels. Figure 5.8 shows that similar results are obtained for both experiments. Indeed, we observe that the sensibility is decreasing while the percentage of voxels considered is decreasing, whereas, the specificity is more stable, but both quantities are more and more variable. Taking less than 1% of the voxels in our algorithm leads to results with too high variability: we cannot accept that different execution (with random voxel selections) lead to different results.

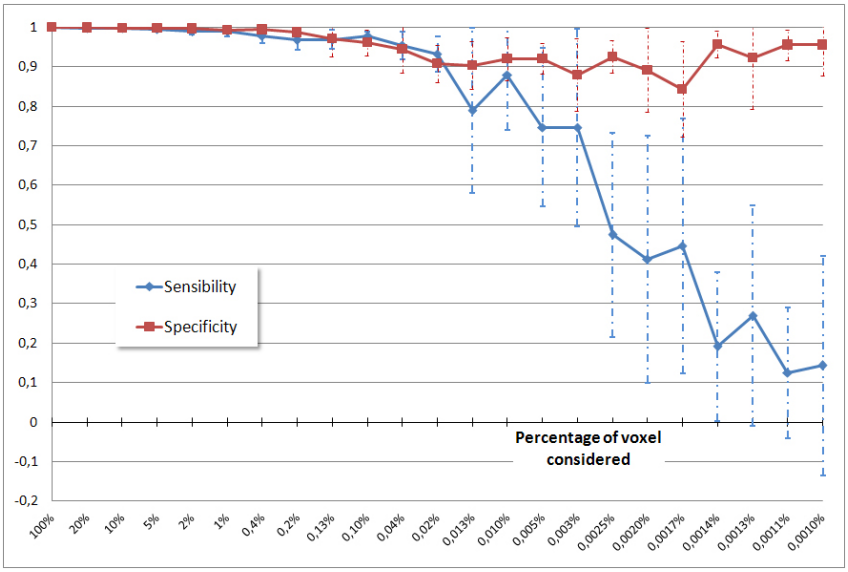
Specificity: A WM segmentation with a specificity of 100% would mean that each voxel defined as belonging to (resp. not to) the white matter is really belonging to (resp. not to) the white matter in the segmentation of reference. So we could say that decreasing the number of voxel considered for the method doesn't affect the results (without taking into account the variability). But this doesn't mean that our segmentation results are accurate for low percentage of voxels. Indeed, in our case, specificity and accuracy should not be confused, because there are far more true negatives (voxels out of brain) than true positives (voxels really belonging to WM).

Sensibility: The drastic decrease of the sensibility describes an increase of the number of false negative which correspond to the voxel really belonging to the WM but not labelled as such. This reveals that after a certain threshold value of the percentage of voxels, the number of voxel is not any more sufficient for the EM method to be able to define the Gaussian class parameter.

As a conclusion we can say that these results reveal that using only 1% of the voxels of the image in the Expectation Maximization method would divide its execution time by 3 or 4 (compared to the execution with 100% of the voxels), without impacting the WM segmentation quality (*C.f.* Figure 5.9).

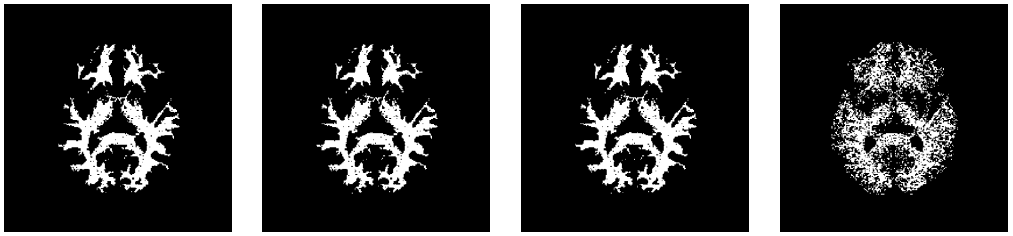


(a) Normal subject



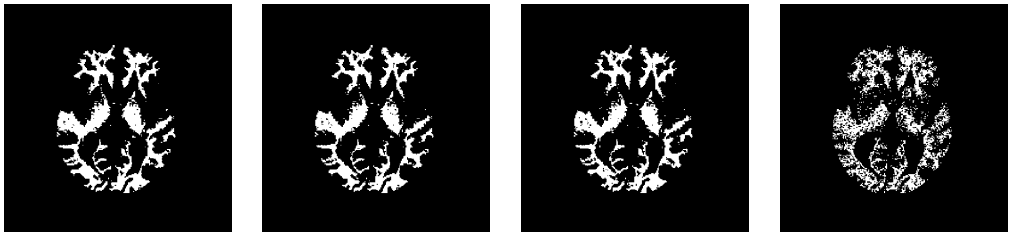
(b) Patient affected by Relapsing/Remitting Multiple Sclerosis

FIGURE 5.8 - *Mean sensibility and specificity of white matter segmentations in function of the percentage of voxel considered to compute the parameters of the brain compartments, and their variations. Each point corresponds to a mean of 9 executions (where the same number of voxels are randomly chosen).*



(a) (b) (c) (d)

Segmentation from the normal subject



(a) (b) (c) (d)

Segmentation from the patient affected by Multiple Sclerosis

FIGURE 5.9 - ***White matter binary segmentations generated with the workflow for different percentage of voxel considered : a) 100%, b) 1%, c) 0.1%, d) 0.005%.***

Chapter 6

Conclusion

In this report, a description of a brain segmentation method into healthy compartments classes and its deployment on the EGEE grid has been presented.

To become familiar with this pipeline, development of tools in the software SepINRIA has been made. SepINRIA is dedicated to Multiple Sclerosis and based on the same algorithms. The implementation of the new tools, that is to say, an automatic brain atrophy evaluation method as well as a DICOM export and a display of images by fusion, leads to a new release of the software. Thus, work on SepINRIA has been recognized once again useful for clinical research and application, since an article (abstract and poster format) on which I am the first author has been accepted. This article: "SepINRIA v1.7.2: Multiple Sclerosis Brain MRI visualization, comparison and analysis Software" [Pernod 2008a], has been submitted and accepted for the Montréal 2008 World Congress on Treatment and Research in Multiple Sclerosis (joint meeting of ACTRIMS, ECTRIMS and LACTRIMS). A version of the poster is available in Appendix A.6.

The problematic of this work was to understand the step needed to deploy a real application on a grid. Thus, a complete investigation of the tools made available or simply used by the I3S Laboratory has been done. Besides experiments demonstrate that this application of brain segmentation is well adapted to grid and provide a sizeable gain of time in multiple executions. Results of the workflow have been confronted to local results and have been successfully validated.

This work leads me to the writting of a full article for the MICCAI 2008 grid workshop in New York which has been accepted: "Multiple Sclerosis Brain MRI Segmentation Workflow deployment on the EGEE Grid" [Pernod 2008b]. This article will be presented in September by a researcher of Asclepios.

For that occasion, research have been done to test the influence of a parameter our method of segmentation in order to improve the algorithm speed. Our main finding is that in the expectation-maximization algorithm, taking only a part of the voxels doesn't affect severely the estimation of the Gaussian class parameter until a critical value. Thus, if needed, the brain healthy compartments classes could be generated faster while keeping a good accuracy. But this experiment also highlights the grid issue of bottleneck effect. In future work, testing different gLite parameters to increase the performance of workload is needed.

In the framework of the NeuroLOG project, an other article submitted by I3S and referring to our application (abstract and poster) has been accepted for the 2008 Enabling Grid for E-sciencE (EGEE) conference: "NeuroLOG: neuroscience application workflows execution on the

EGEE grid" [Rojas Balderrama 2008].

Finally, seven services of our workflow have been selected to be shared to all partner of the NeuroLOG community. Generalization of the codes (in order to take different images format) and full description have been done. The interest being that each service is an independent block and can be added to different workflows. Thus, this allows either to create new workflow with them or to test algorithms, by changing only one block in a workflow for example.

In the point of view of my experience, this intership was very interesting and introduce me to new knowledges, as well as in image processing as in grid computing. I also appreciate the fact that this work was part of a community adding to my experience work in group skills like communication to hand on knowledge from one partner to others.

Appendix A

Figures

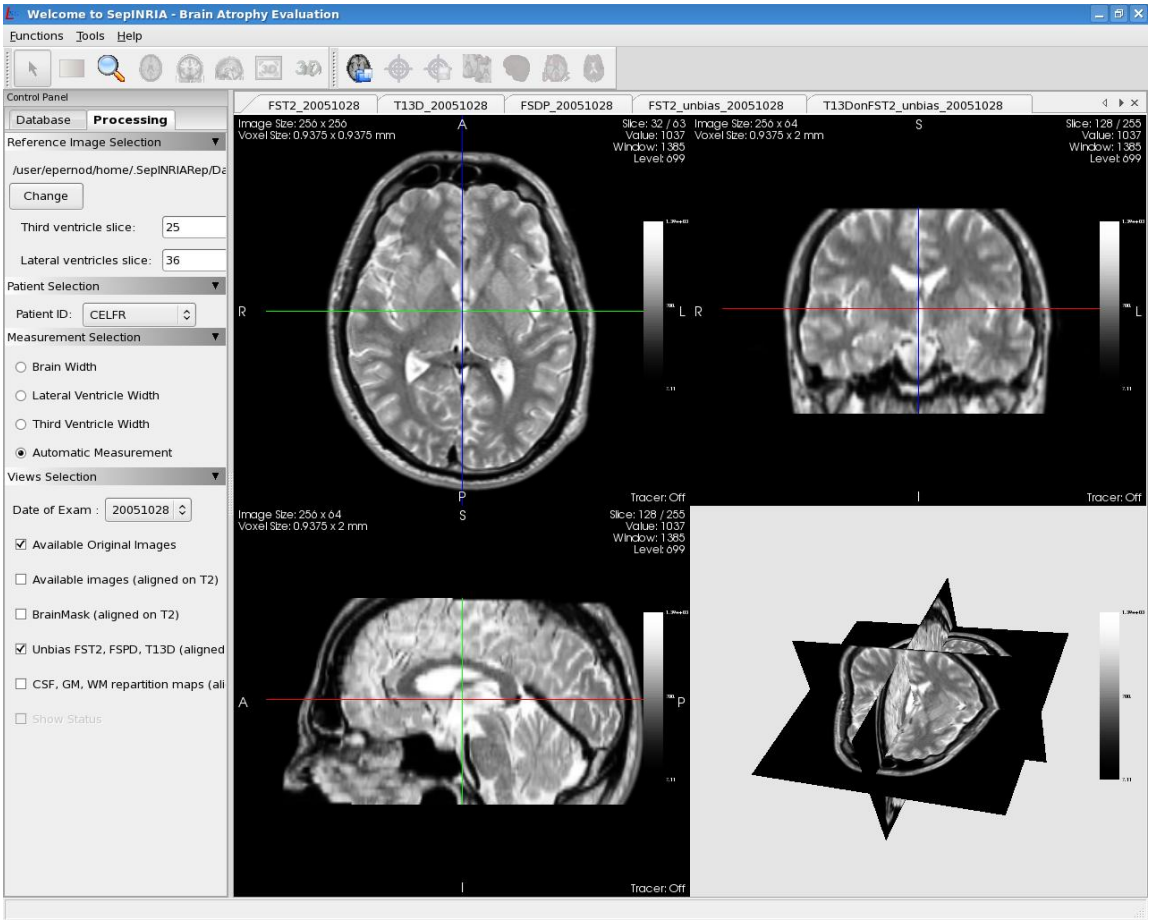


Figure A.1 - *SepINRIA software, Brain Atrophy Functionality screenshot. Display of the three views of a 3D patient MRI sequence and its 3D view. Process panel belonging to this functionality can be observed on the left side. Display is done using VTK and vtkINRIA3D libraries while the interface is based on wxWidgets.*

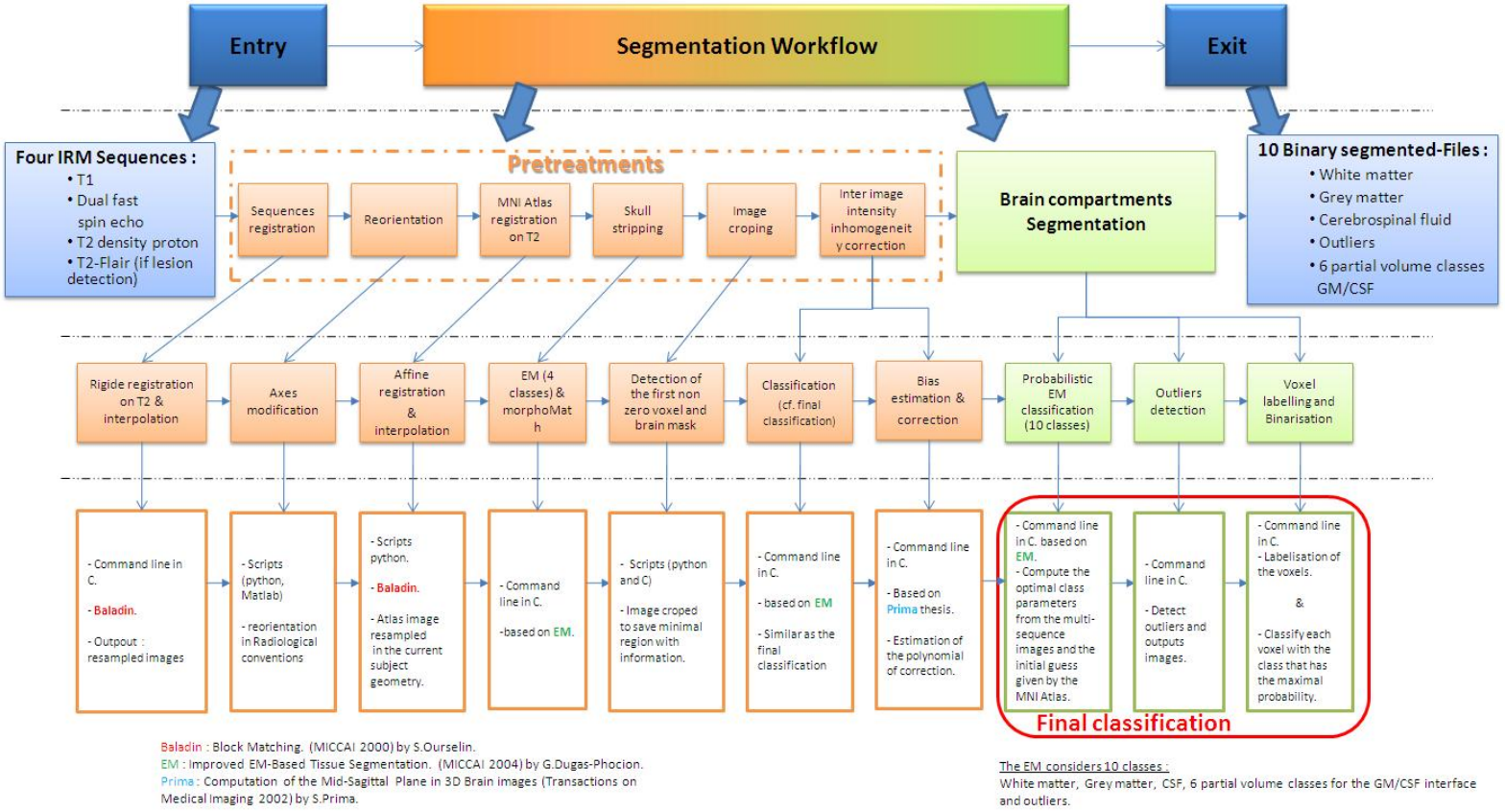


FIGURE A.2 - *Complete description of the local pipeline of brain MRI segmentation. Different level of description are available, from the global algorithms name to the used code language. Even if this description is complete, possibility of parallelism is not express in this scheme.*

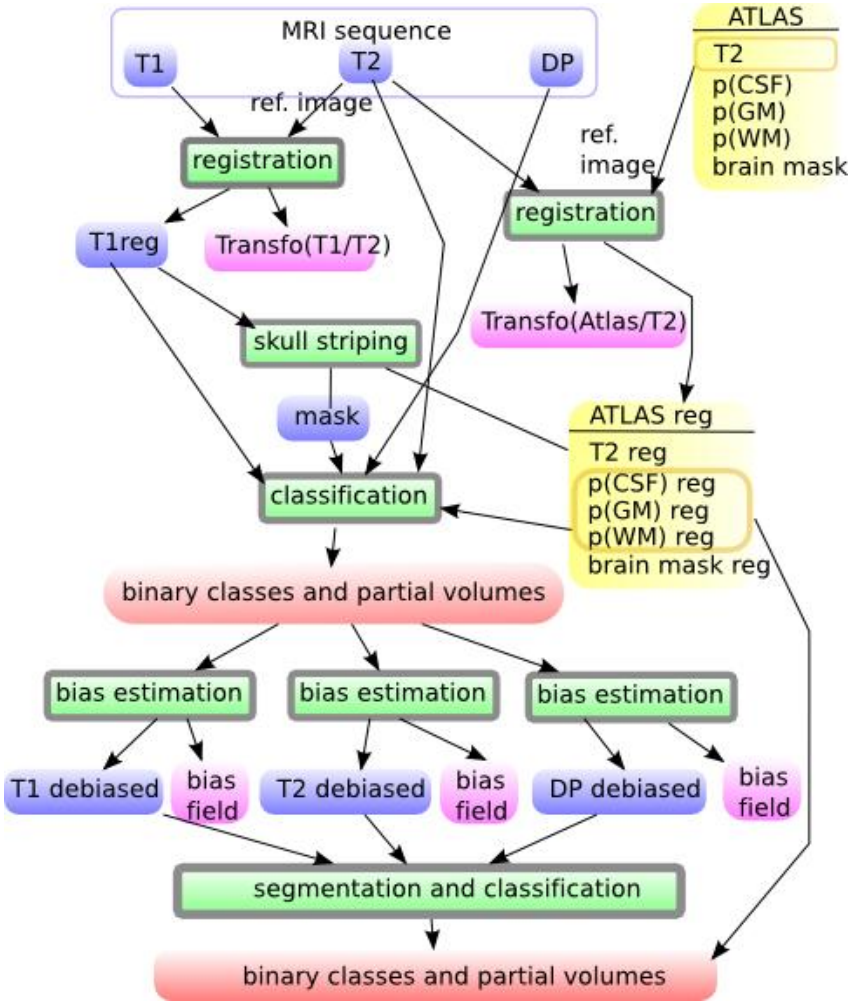


FIGURE A.3 - *Second description of the pipeline based on the step which can be isolated as black boxes (in green). Temporally results are represented in purple, inputs in blue, MNI Atlas in yellow and m;portant results in red. This scheme allows to imagine parallelism possibilities.*

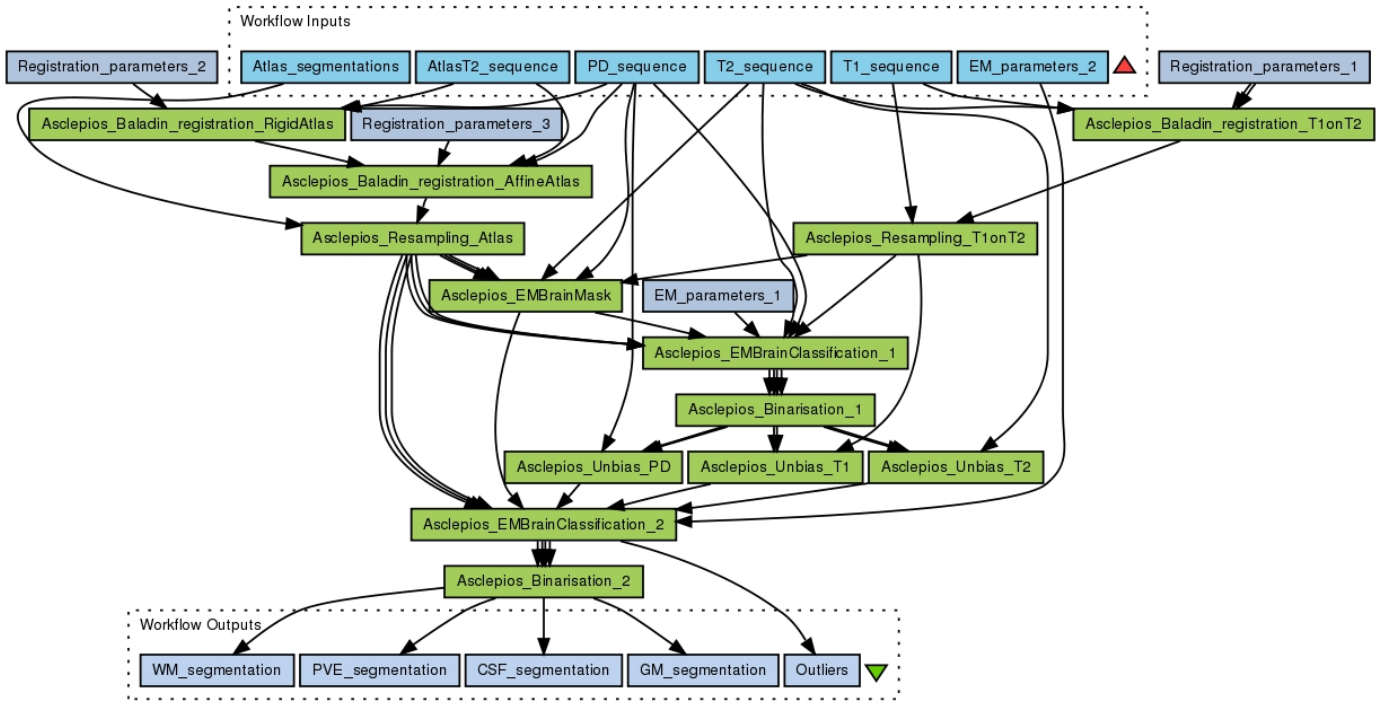


FIGURE A.4 - *Graphical view of the workflow generated by the software Taverna. In fact, the file is in Scufl language. Inputs, outputs and services in green (also called processor in Taverna) are described and organized. However this scheme is simplified. Indeed, it does not display the port*

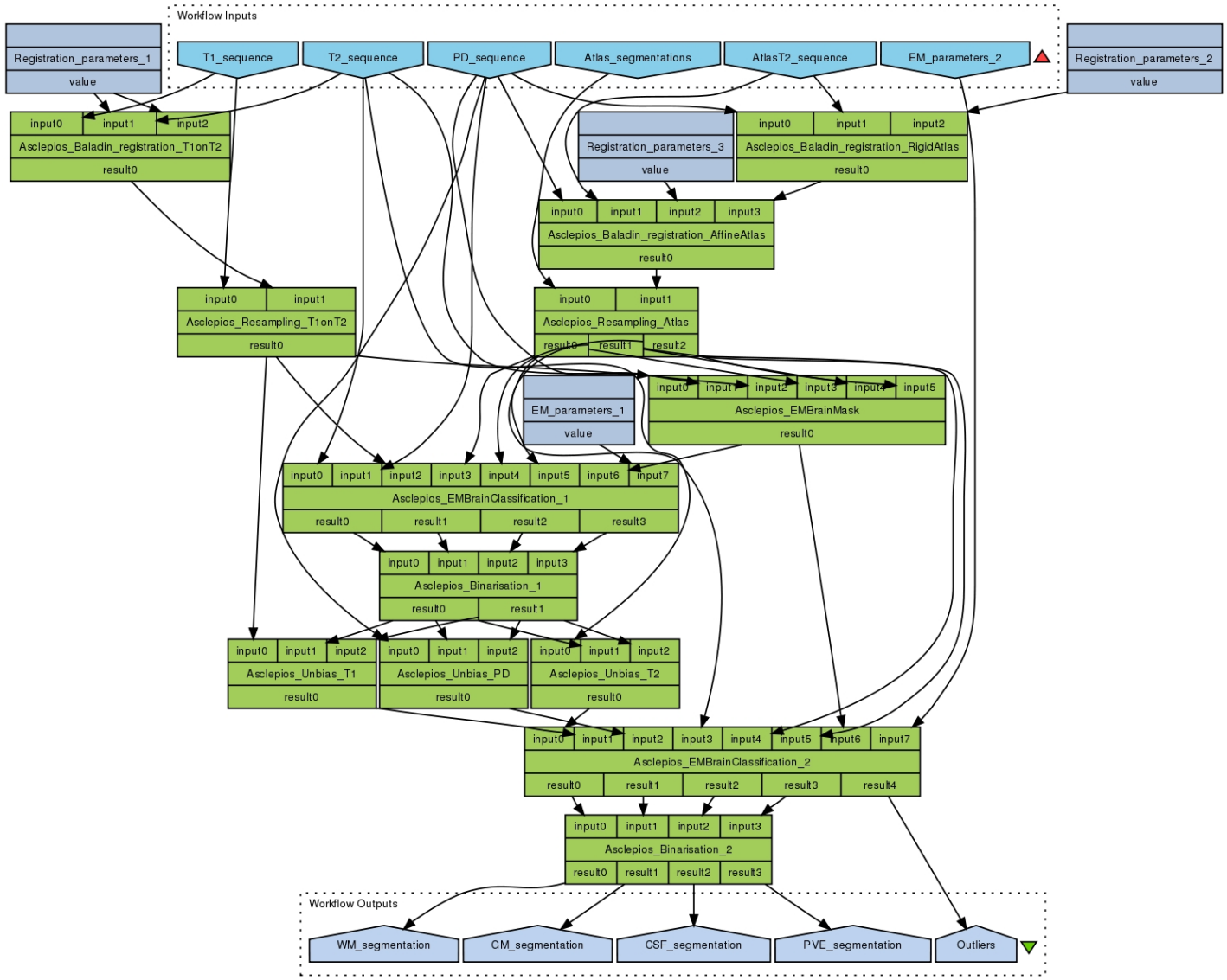


FIGURE A.5 - *Same graphic as Figure A.4 but adding the service ports view.*

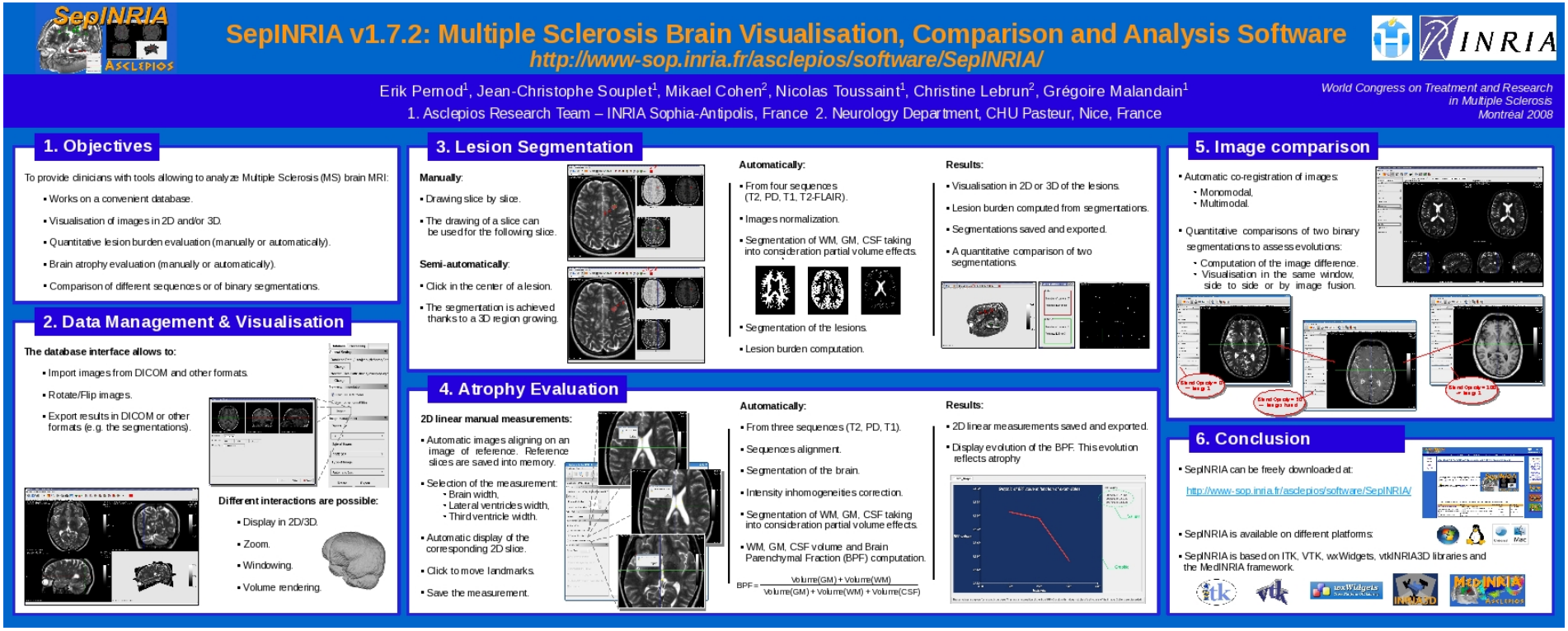


FIGURE A.6 - *Poster on the software SepINRIA: "SepINRIA v1.7.2: Multiple Sclerosis Brain MRI visualization, comparison and analysis Software".. For the Montréal 2008 World Congress on Treatment and Research in Multiple Sclerosis (joint meeting of ACTRIMS, ECTRIMS and LACTRIMS).*

Bibliography

[Charcot 1898] J.M. Charcot. Leçons sur les maladies du système nerveux, volume 1. Bourneville, 1898. 6

[Dugas-Phocion 2004] Guillaume Dugas-Phocion, Miguel Ángel González Ballester, Grégoire Malandain, Christine Lebrun et Nicholas Ayache. *Improved EM-Based Tissue Segmentation and Partial Volume Effect Quantification in Multi-Sequence Brain MRI.* In Proc. of MICCAI'04, volume 3216 of *Lecture Notes in Computer Science*, pages 26–33, Saint-Malo, France, September 2004. Springer. 14

[Dugas-Phocion 2006] Guillaume Dugas-Phocion. *Segmentation d'IRM Cérébrales Multi-Séquences et Application à la Sclérose en Plaques.* PhD thesis, École des Mines de Paris, March 2006. 11, 14

[Evans 1992] A.C. Evans, D.L. Collins, P. Neelin, D. MacDonald, M. Kamber et T.S. Marrett. *An MRI-based stereotactic brain atlas from 300 young normal subjects.* In In 22nd Symposium of the Society for Neuroscience, page 408, Anaheim, 1992. 12

[Foster 2001] Ian Foster, Carl Kesselman et Steven Tuecke. *The Anatomy of the Grid: Enabling Scalable Virtual Organizations.* International Jounral of Supercomputer Applications, vol. 15, no. 3, 2001. 25

[Glatard 2006a] Tristan Glatard, David Emsellem et Johan Montagnat. *Generic web service wrapper for efficient embedding of legacy codes in service-based workflows.* In Grid-Enabling Legacy Applications and Supporting End Users Workshop (GELA'06), Paris, France, June 19-23, 2006. 28

[Glatard 2006b] Tristan Glatard, Johan Montagnat et Xavier Pennec. *Efficient services composition for grid-enabled data-intensive applications.* In Proceedings of the IEEE International Symposium on High Performance Distributed Computing (HPDC'06), Paris, France, June 19, 2006. 30

[Glatard 2007] Tristan Glatard. *Description, deployment and optimization of medical image analysis workflows on production grids.* Thèse de sciences (phd thesis), Université de Nice – Sophia-Antipolis, November 2007. 29

[Hill 2001] D. L. Hill, P. G. Batchelor, M. Holden et D. J. Hawkes. *Medical image registration.* Phys Med Biol, vol. 46, no. 3, March 2001. 12

[Lauterbur 1973] P.C Lauterbur. *Image formation by induced local interactions: examples employing nuclear magnetic resonance.* Nature, vol. 242, pages 190–191, 1973. 8

[Liang 1999] Z.P. Liang et P.C. Lauterbur. Principles of Magnetic Resonance Imaging. BIOMEDICAL ENGINEERING. IEEE PRESS, 1999. 8

[Lingrand 2008] Diane Lingrand, Johan Montagnat et Tristan Glatard. *Estimating the execution context for refining submission strategies on production grids.* In Workshop ASSESS / Modern BIO (CCgrid'08), pages 753 – 758, Lyon, May 2008. IEEE. 38

[McIlroy 1969] M. D. McIlroy. *"Mass Produced" Software Components.* In P. Naur et B. Randell, editors, Software Engineering, pages 138–155, Brussels, 1969. Scientific Affairs Division, NATO. Report of a conference sponsored by the NATO Science Committee, Garmisch, Germany, 7th to 11th October 1968. 27

[Montagnat 2006] Johan Montagnat, Tristan Glatard et Diane Lingrand. *Data composition patterns in service-based workflows.* In Workshop on Workflows in Support of Large-Scale Science (WORKS'06), Paris, France, June 2006. 33

[Montagnat 2008] Johan Montagnat, Alban Gaignard, Diane Lingrand, Javier Rojas Balderrama, Philippe Collet et Philippe Lahire. *NeuroLOG: a community-driven middleware design.* In HealthGrid, Chicago, June 2008. IOS Press. 4

[Oinn 2004] Tom Oinn, Matthew Addis, Justin Ferris, Darren Marvin, Martin Senger, Mark Greenwood, Tim Carver, Kevin Glover, Matthew R. Pocock, Anil Wipat et Peter Li. *Taverna: a tool for the composition and enactment of bioinformatics workflows.* Bioinformatics, vol. 20, no. 17, pages 3045–3054, 2004. 30

[Ourselin 2000] S. Ourselin, A. Roche, S. Prima et N. Ayache. *Block Matching: A General Framework to Improve Robustness of Rigid Registration of Medical Images.* In A.M. DiGioia et S. Delp, editors, Third International Conference on Medical Robotics, Imaging And Computer Assisted Surgery (MICCAI 2000), volume 1935 of *Lectures Notes in Computer Science*, pages 557–566, Pittsburgh, Penn, USA, octobre 11-14 2000. Springer. 12

[Pernod 2008a] Erik Pernod, Jean-Christophe Souplet, Mikael Cohen, Nicolas Toussaint, Christine Lebrun et Grégoire Malandain. *SepINRIA v1.7.2: Multiple Sclerosis Brain MRI visualisation, comparison and analysis Software.* In World Congress for Treatment and Research in Multiple Sclerosis (WCTRIMS), Montreals, Canada, September 2008. . 43

[Pernod 2008b] Erik Pernod, Jean-Christophe Souplet, Javier Rojas Balderrama, Diane Lingrand et Xavier Pennec. *Multiple Sclerosis Brain MRI Segmentation Workflow deployment on the EGEE Grid.* In MICCAI-Grid Workshop (MICCAI-Grid), New York, NY, USA, September 2008. 43

[Polman 2005] Chris H H. Polman, Stephen C C. Reingold, Gilles Edan, Massimo Filippi, Hans-Peter P. Hartung, Ludwig Kappos, Fred D D. Lublin, Luanne M M. Metz, Henry F F. McFarland, Paul W W. O'connor, Magnhild Sandberg-Wollheim, Alan J J. Thompson, Brian G G. Weinshenker et Jerry S S. Wolinsky. *Diagnostic criteria for multiple sclerosis: 2005 revisions to the "McDonald Criteria".* Ann Neurol, November 2005. 7

[Prima 2001] S. Prima, N. Ayache, T. Barrick et N. Roberts. *Maximum Likelihood Estimation of the Bias Field in MR Brain Images: Investigating Different Modelings of the Imaging*

Process. In W.J. Niessen et M.A. Viergever, editors, 4th Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI'01), volume LNCS 2208, pages 811–819, Utrecht, The Netherlands, Oct. 2001. 13

[Rojas Balderrama 2008] Javier Rojas Balderrama, Diane Lingrand, Erik Pernod, Jean-Christophe Souplet, Xavier Pennec et Johan Montagnat. *NeuroLOG: neuroscience application workflows execution on the EGEE grid.* In Enabling Grid for E-sciencE (EGEE) conference, Istanbul, Turkey, September 2008. . 44

[Sled 1998] J.G. Sled, A.P. Zijdenbos et A.C. Evans. *A Nonparametric Method for Automatic Correction of Intensity Nonuniformity in MRI Data.* IEEE Trans Med Imaging, vol. 17, no. 1, pages 87–97, 1998. 13

[Souplet 2007] Jean-Christophe Souplet, Christine Lebrun, Pierre Clavelou, William Camu, Stéphane Chanalet, Nicholas Ayache et Grégoire Malandain. *A comparative study of skull stripping methods in relapsing-remitting multiple sclerosis: Emergence of a new automatic segmentation algorithm.* In Congress of the European Committee for Treatment and Research in Multiple Sclerosis (ECTRIMS), Prague, Czech Republic, October 2007. 12

[Souplet 2008a] Jean-Christophe Souplet, Christine Lebrun, Nicholas Ayache et Grégoire Malandain. *An Automatic Segmentation of T2-FLAIR Multiple Sclerosis Lesions.* In MICCAI-Multiple Sclerosis Lesion Segmentation Challenge Workshop, New York, NY, USA, September 2008. 16

[Souplet 2008b] Jean-Christophe Souplet, Christine Lebrun, Nicholas Ayache et Grégoire Malandain. *A New Evaluation Of The Brain Parenchymal Fraction: Application In Multiple Sclerosis Longitudinal Studies.* In Proceedings of the IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI'08), pages 65–68, Paris, France, May 2008. IEEE. 16

[Souplet 2008c] Jean-Christophe Souplet, Christine Lebrun, Stéphane Chanalet, Nicholas Ayache et Grégoire Malandain. *Revue des approches de segmentation des lésions de sclérose en plaques dans les séquences conventionnelles IRM.* Revue Neurologique, 2008. Accepted for publication. 10, 15

[Yu 2005] Jia Yu et Rajkumar Buyya. *A taxonomy of scientific workflow systems for grid computing.* SIGMOD Rec., vol. 34, no. 3, pages 44–49, 2005. 29