A Horizontal Data Fusion Toolbox: Putting the focus on interoperability

B. Bleuzé, M. Sermesant, J. Olivain, V. Garcia, N. Toussaint*
Asclepios Team, Inria Sophia

1 Introduction

The Virtual Physiological Human (VPH) research aims at integrating in computational models an extensive representation of the human body. Given the numerous phenomena interacting in such a complex living system, several different modalities of observation are needed in order to gather information on these phenomena. Moreover following up on the time evolution implies comparing data at different points in time. Finally, superposition of simulated volumes and acquired data is also a needed tool. All these needs require a so called "Horizontal Data Fusion" which registers in the same spatial coordinates different datasets.

However, pairing the numerous image modalities types and file formats around with the many registration algorithms is far from being straight forward. Indeed, each registration algorithm has its own input/output formats, graphical user interface and visualisation. Thus, its use by a non-expert is challenging and the combination or comparison of such algorithms is difficult.

This problem has been treated closely in the Tool-kit Content Guidelines(1), published by the VPH Network of Excellence[1], specifically in the Interoperability Document. The aim of our project at Inria within the VPH is to provide a toolbox enabling the visualisation and the aggregation of data fusion tools and algorithms, being one of the first real-life implementations following these guidelines.

This work is included in *medInria* v2[2]. Through the use of a core/plug-ins architecture, it explores solutions to make it easy for researchers to import their own tools, wrap them around a single application or exchange them between applications supporting the same framework. That attempt at a universal solution is a step towards standardisation of the exchange of algorithms. Standardisation calls for sustainability, maintainability. The proposed framework can also help quantitative and qualitative tests of functionally similar methods.

medInria v2 implements registration algorithms, using an API for plug-ins we propose as an open framework and an open source registration programming interface. This software allows the visualisation of several formats of images, all present in the Data Characterisation Guidelines(1).

2 Architecture

medInria v2 demonstrates the viability of a core/plug-in architecture developed at Inria in the dtk project[3]. The visualisation code takes advantage of the work done in earlier versions of *MedINRIA* 1.x (2) and vtkinria3d (3).

Figure 1 describes the general architecture. The kernel (or core) provides an abstraction layer allowing the interaction between data, processes and views. A plug-in is an implementation of a data reader, writer, process or view. Registration algorithms are typical processes. Data are inputs and outputs for processes, views consume Data as inputs as well. They use data such as itk image objects loaded for instance from DICOM files through readers. Views can be a VTK view for instance, or any other viewer. Object factories take care of the instantiation. A typical registration algorithm takes 2 data inputs: a fixed image that will act as a reference, and a so called moving image on which the transformation will be applied. The output is a third data object: the product of the transformation. The plug-ins are interchangeable, and an affine transformation can be replaced with a non linear one by just switching the process objects, which are just dynamic libraries, without any change to the rest of the software.

---

*N. Toussaint is now with the King's College London, Division of Imaging Sciences, St Thomas' Hospital

[1] http://www.vph-noe.eu/

[2] http://med.inria.fr

[3] http://www.dtk.inria.fr

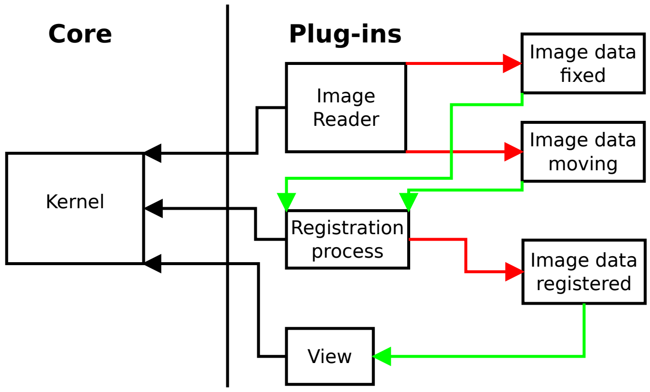

Figure 1: General Architecture for registration algorithms. (inputs are green, output red)

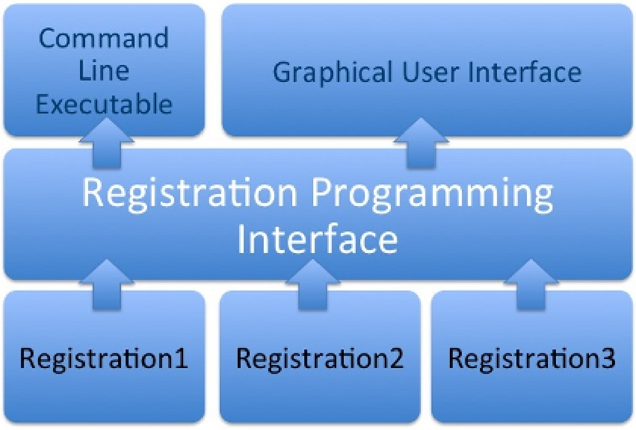

Figure 2: Registration Programming Interface

A clear advantage of that abstraction layer delivered as an open API is that open and close source plug-ins from different institutions or vendors can live together in the same execution environment without any trouble, as long as the licences allow linking to other programmes.

As for registration plug-ins themselves, all the implemented algorithms use the Registration Programming Interface (RPI)[4]. This API defines the common properties of a registration work-flow: the two inputs, the output, a transformation (matrix, displacement vector field or velocity field). It also brings some tools for re-sampling the output image, observing the progression of the algorithm, and more. Specific implementations only add methods to access and set their parameters.

Wrapping all registration algorithms with RPI allows for very fast integration in larger software projects, as highlighted in figure 2.

3 Demonstrator

medInria has an intuitive and slick looking interface, that makes it easy to use, and also particularly appealing to clinicians thanks to its simplicity3. It is organised in workspaces, one of them taking care of the Registration work-flow. The programme imports and stores DICOM, nifti and other files, which are then used by the workspaces. Each process run from a workspace generate data (itk volumes in the registration's case), that can then be exported to many standard formats. Having the full *medInria* software above the registration workspace allows us to reach the second level of interoperability: the Syntactic level, as described in VPH Guidelines. We can exchange data with most standard formats.

So far we have added five algorithms from the Inria Asclepios team, the multi-modal diffeomorphic demons(4) developed by the ISTB of Bern is being included, and talks about the addition of the NiftyReg(5) algorithm is also in discussion.

4 Perspectives

Within the VPH Tool-kit, the objective of such toolbox is to aggregate the different tools existing in the community, thus it should suit the needs of other already existing algorithms to be wrapped as *medInria* plug-ins. Moreover, by releasing a simple API and the software, the objective is also to attract more developers to the framework, so that any researcher can create his own plug-in. Interoperability which is only done now through the use of standard data format, and an open API (including the RPI), is likely to be extended even further by the addition of a python scripting engine, to chain whole work-flows of processing tools, and maybe even the DICOM part 19 Application Hosting standard[5].

[4] http://www-sop.inria.fr/asclepios/software/RPI/
[5] ftp://medical.nema.org/medical/dicom/2011/11_19pu.pdf

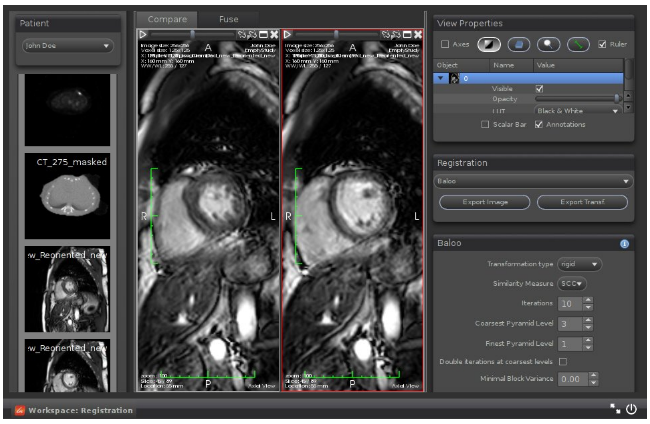

Figure 3: *medInria* v2: the registration workspace

5 Conclusion

Combining the large number of file formats supported by *medInria*, its open framework for writing plug-ins, and the Registration Programming Interface has proven to be a very promising recipe for aggregating more algorithms. Full semantic interoperability between different platforms always require either wrapping existing code or choose the most widely used framework. Only following strict guidelines will help reaching interoperability between a large number of frameworks, which only then could establish themselves as a standard.

References

[1] members NoE VPH. VPH Content Toolkit Guidelines; 2011. http://toolkit.vph-noe.eu/toolkit-guidelines.

[2] Toussaint N, Souplet JC, Fillard P. MedINRIA: Medical Image Navigation and Research Tool by INRIA. In: Proc. of MICCAI'07 Workshop on Interaction in medical image analysis and visualization. Brisbane, Australia; 2007. .

[3] Toussaint N, Sermesant M, Fillard P. vtkINRIA3D: A VTK Extension for Spatiotemporal Data Synchronization, Visualization and Management. In: Proc. of Workshop on Open Source and Open Data for MICCAI. Brisbane, Australia; 2007. .

[4] Lu H, Reyes M, Serifovic A, Weber S, Sakurai Y, Yamagata H, et al. Multi-modal diffeomorphic demons registration based on point-wise mutual information. In: Biomedical Imaging: From Nano to Macro, 2010 IEEE International Symposium. Rotterdam, The Netherlands; 2010. .

[5] Modat M, Ridgway GR, Taylor ZA, Lehmann M, Barnes J, Hawkes DJ, et al. Fast free-form deformation using graphics processing units. Computer Methods and Programs in Biomedicine. 2010;.