1st MICCAI Workshop on

MAPPING2015

MAnagement and Processing of images for Population ImagiNG

Proceedings

Editors

Christian Barillot

Michel Dojat

David Kennedy

Wiro Niessen

MICCAI-MAPPINGI2015 Oct 5, 2015 Munich, Germany

https://project.inria.fr/fli/mapping-workshop/

<Authors>, <Paper Title>, In: *Proceedings of the 1st Miccai 2015 Workshop on Management and Processing of images for Population Imaging – MICCAI-MAPPING2015*, C . Barillot, M. Dojat, D. Kennedy and W. Niessen (Eds), pp. <Page Numbers>, 2015.

Table of Contents

Preface **5**

Workshop Organisation **6**

Data-driven probabilistic atlases capture whole-brain individual variation *Yuankai Huo, Katherine Swett, Susan Resnick, Laurie Cutting, Bennett Landman* **7**

A neuroscience gateway for handling and processing population imaging studies M.*W.A. Caan, J. Teeuw, S. Shahand, M. M. Jaghoori, J. Huguet, A. van Altena, S.D. Olabarriaga* **15**

Shanoir: Software as a Service Environment to Manage Population Imaging Research Repositories *Christian Barillot, Elise Bannier, Olivier Commowick, Isabelle Corouge, Justine Guillaumont, Yao Yao , Michael Kain* ..**23**

Population Imaging Study IT Infrastructure: An Automated Continuous Workflow Approach *Marcel Koek, Hakim Achterberg, Marius de Groot, Erwin Vast, Stefan Klein, Wiro Niessen* **31**

Fastr: a workflow engine for advanced data flows *Hakim Achterberg, Marcel Koek, Wiro Niessen* **39**

Design and implementation of a generic DICOM archive for clinical and pre-clinical research *Julien Lamy, Romain Lahaxe, Jean-Paul Armspach, Fabrice Heitz* **47**

Preface

Several recent works underline methodological points that limit the validity of published results, for instance in neuroimaging studies (Button et al Nat Rev Neurosc 14:1-12, 2013, Ionnadis et al. TICS 1-7, 2014, Carp J NeuroIm 63:289-300, 2012). One of the themes is the endemic low statistical power of the published studies due to the small size of population involved. To overcome this aspect cohort studies should be promoted. This workshop was dedicated to the methodological aspects and solutions to support the constitution, the management and the processing of such large cohorts and their link to image processing infrastructures for the sharing and execution of processing workflows through software and hardware architectures. This encompasses the aspects of application ontologies, data structures, new paradigms for handing data, interoperability of repositories, semantic queries, image processing composition, machine learning, data mining and high performance computing,... and the pros and cons aspects of existing working solutions.

This proceedings addresses the *Methodological Issues for Population Imaging* related to the topics of data management and processing of large imaging data bases, including Infrastructure for facilitating data and software sharing and reused; Conceptual and technical methods for solving specific difficult points (domain ontology development, image processing pipeline development, grid access facilitation, data mining and machine learning on big data ...); Case studies using specific platforms (pros and cons, ...), and needs and requirements for specific multi-centre studies.

The workshop was partially supported by the *France Life Imaging* national project in France (ANR-11-INBS-006). It was organized in an half day event during the MICCAI 2015 conference in Munich, Germany. It with two invited speakers:

- Prof. **Monique Breteler**, Director of Population Health Sciences, German Center for Neurodegenerative Diseases (DZNE) , Helmholtz, Bonn, Germany
- Prof. **Gunter Schumann**, Chair in Biological Psychiatry, MRC-SGDP Centre, Institute of Psychiatry, King's College, London, UK

Christian Barillot
Michel Dojat
David Kennedy
Wiro Niessen

Workshop Organization

Workshop Chairs:

Christian Barillot	*CNRS, Rennes, France*
Michel Dojat	*Inserm, Grenoble France*
David Kennedy	*Univ. Massachusetts, Boston, MA, USA*
Wiro Niessen	*Erasmus Univ., Rotterdam, The Netherlands*

Program Committee

Alan Evans	U*niv. McGill, Montreal, Canada*
Bernard Mazoyer	*CNRS, Bordeaux, France*
Bertrand Thirion	*Inria, Saclay Center, France*
Camille Maumet	*Warwick University, Coventry, UK*
Daniel Rueckert	*Imperial College, London, UK*
Krzysztof Gorgolewski	*Stanford Univ., CA, USA*
Jean-Baptiste Poline	*CEA Neurospin, Saclay, France*
Johan Montagnat	*CNRS, Sophia-Antipolis, France*
Meike Vernooij	*Erasmus Univ., Rotterdam, The Netherlands*
Russ A. Poldrack	*Stanford Univ., CA, USA*
Silvia Delgado Olabarriaga	*University of Amsterdam, The Netherlands*
Susan Shenkin	*University of Edinburgh, UK*
Tom Nichols	*Warwick University, Coventry, UK*
Tristan Glatard	*CNRS, Lyon, France*

Data-driven Probabilistic Atlases Capture Whole-brain Individual Variation

Yuankai Huo[1], Katherine Swett[2], Susan M. Resnick[3], Laurie E. Cutting[2], Bennett A. Landman[1]

[1] Electrical Engineering, Vanderbilt University, Nashville, TN, USA
[2] Special Education, Vanderbilt University, Nashville, TN, USA
[3] National Institute on Aging, Baltimore, MD, United States

Abstract. Probabilistic atlases provide essential spatial contextual information for image interpretation, Bayesian modeling, and algorithmic processing. Such atlases are typically constructed by grouping subjects with similar demographic information. Importantly, use of the same scanner minimizes inter-group variability. However, generalizability and spatial specificity of such approaches is more limited than one might like. Inspired by Commowick's "Frankenstein's creature paradigm" which builds a personal specific anatomical atlas, we propose a data-driven framework to build a personal specific probabilistic atlas under the large-scale data scheme. The data-driven framework clusters regions with similar features using a point distribution model to learn different anatomical phenotypes. Regional structural atlases and corresponding regional probabilistic atlases are used as indices and targets in the dictionary. By indexing the dictionary, the whole brain probabilistic atlases adapt to each new subject quickly and can be used as spatial priors for visualization and processing. The novelties of this approach are (1) it provides a new perspective of generating personal specific whole brain probabilistic atlases (132 regions) under data-driven scheme across sites. (2) The framework employs the large amount of heterogeneous data (2349 images). (3) The proposed framework achieves low computational cost since only one affine registration and Pearson correlation operation are required for a new subject. Compared with site-based group atlases, the experimental results show that the proposed atlases capture more individual variations by decreasing the Jensen–Shannon divergence between probabilistic atlases and the ground truth. Our method matches individual regions better with higher Dice similarity value when testing the probabilistic atlases. Importantly, the advantage the large-scale scheme is demonstrated by the better performance of using large-scale training data (1888 images) than smaller training set (720 images).

Keywords: Atlas, Data Mining, Clustering, Data-Driven, Large-scale Data

1 Introduction

Probabilistic atlases play important roles in understanding the spatial variation of brain anatomy, in visualization, and in the processing of data. The basic framework of making probabilistic atlases is to bring the image data from the selected subjects into an atlas space by rigid or non-rigid registration [1]. Then, probabilistic maps are gen-

Yuankai Huo, Katherine Swett, Susan Resnick, Laurie Cutting, Bennett Landman ; Data-driven probabilistic atlases capture whole-brain individual variation, In: Proceedings of the 1st Miccai 2015 Workshop on Management and Processing of images for Population Imaging – MICCAI-MAPPING2015, C . Barillot, M. Dojat, D. Kennedy and W. Niessen (Eds), pp.7-14, 2015.

erated by averaging the segmentations of regions from a specific group of subjects with similar demographic data, such as age, sex and from the same site. However, the inter-subject variability is normally larger than the inter-group variability, which causes the group-based scheme to fail to capture a great deal of individual variation.

To overcome the large inter-subject variability, Commowick et al. proposed the "Frankenstein's creature paradigm" to build a personal specific anatomical atlas for head and neck region [2]. The paradigm first selected regional anatomical atlases based on a training database then merged them together into a complete atlas. However, this framework cannot be directly applied on making probabilistic atlases since each probabilistic atlas is averaged from a group of segmentations. Moreover, compared with the 105 CT images used as the database in Commowick's framework, we employ 2349 heterogeneous MRI images in our framework.

In this paper, we propose a large-scale data-driven framework to learn a dictionary of the whole brain probabilistic atlases (132 regions) from 1888 heterogeneous 3D MRI training images. The novel contributions of this paper are (1) providing a new data-driven perspective of making whole brain probabilistic atlas, (2) generating the more accurate personal specific probabilistic atlases by using the large-scale data from different groups and even different sites, and (3) achieving low computational cost of applying the learned dictionary on new subjects.

2 Data

The dataset aggregates 9 datasets with a total 2349 MRI T1w 3D images obtained from healthy subjects. The 2349 images are divided to 1888 training and 431 testing datasets based on the site and demographic information. The 1888 training images are used to train the data-driven framework ("Training Set 1888"). A subset of 720 training images ("Training Set 720") is employed to generate group atlases (**Table 1**).

Table 1. Data summary of Training Set 720 and Testing Set

	Study	Site	Sex (1 is male)	Age (years)	Scanner (Tesla)	Training (number)	Testing (number)
1	BLSA	NIA	1, 2	29~45	3T	40	0
2	Cutting	Vanderbilt	1, 2	20~30	3T	40	37
3	ABIDE	NYU	1, 2	15~32	3T	40	0
4	IXI	Guys	1	20~45	1.5T	40	22
5	IXI	Guys	2	20~45	1.5T	40	20
6	IXI	HH	1, 2	20~45	3T	40	47
7	IXI	IOP	1, 2	20~45	1.5T	40	0
8	ADHD200	NYU	1, 2	15~17	3T	40	0
9	ADHD200	NeuroIM	1, 2	15~26	3T	40	0
10	ADHD200	Pittsburgh	1, 2	15~20	3T	40	0
11	fcon_1000	Beijing	1	20~26	3T	40	23
12	fcon_1000	Beijing	2	20~26	3T	40	61
13	fcon_1000	Cambridge	1	20~25	3T	40	17
14	fcon_1000	Cambridge	2	21~25	3T	40	39
15	fcon_1000	ICBM	1, 2	19~45	3T	40	0
16	fcon_1000	NewYork	1, 2	20~45	3T	40	52
17	fcon_1000	Oulu	1, 2	20~23	1.5T	40	63
18	NKI_rockland	Rockland	1, 2	15~45	3T	40	35
	OASIS with manual segmentation		1, 2	18~90	3T	0	45
					Total	**720**	**461**

*The Full **Training Set 1888** is obtained from the following datasets:
BLSA: Baltimore Longitudinal Study of Aging
ABIDE: Autism Brain Imaging Data Exchange
ADHD200: Attention Deficit Hyperactivity Disorder
NKI_rockland: Nathan Kline Institute Rockland
NDAR: National Database for Autism Research
Cutting: Data from Cutting pediatric project
IXI: Information eXtraction from Images
fcon_1000: 1000 Functional Connectome
OASIS: Open Access Series on Imaging Study

3 Methods

The proposed data-driven framework consists of two main portions. First, a dictionary is learned by the training data (§3.1-3.3) (**Figure 1**). Second, the learned dictionary is applied to a new subject by affine alignment to MNI space (§3.4-3.5) **(Figure 2)**.

3.1 Get Regional Segmentations and Point Distribution Model

All 720 training subjects were first affinely registered [3] to the MNI305 atlas [4]. Then, a state-of-the-art multi-atlas segmentation (including atlases selection, pairwise registration [5], label fusion [6] and error correction [7]) was performed on each subject. 45 MPRAGE images from OASIS dataset were used as original atlases which are manually labeled with 133 labels (132 brain regions and 1 background) by the BrainCOLOR protocol [8]. Here, we define S_i as the whole brain segmentations with 133 labels and the $i \in \{1,2 \ldots,720\}$ represent different subjects.

Then, a mean segmentation $\bar{S}$ is generated from all $\{S_i\}_{i=1,2,\ldots,720}$ by majority vote label fusion. Since the $\bar{S}$ is smooth, it is a good template of making surface meshes for 132 regions. When the meshes are generated, the vertices $\bar{V}^k$ on the mean segmentation $\bar{S}$ can be propagated to individual segmentations [9]. We non-rigidly register each S_i to $\bar{S}$ and get the diffeomorphism $\phi_i(\cdot)$ [5]. The inverse transformation $\phi_i^{-1}(\cdot)$ is used to propagate the $\bar{V}^k$ back to individual vertices V_i^k (Figure 1).

3.2 Clustering

The Affinity Propagation (AP) clustering method [10] was used to cluster the similar segmentations by using the V_i^k as features. The advantage of AP clustering is it can adaptively cluster the samples into a number of clusters without providing the number of clusters. For region k, the negative mean Euclidian distance $d^k(i,j)$ between vertices V_i^k and V_j^k is used as the similarity measurement for AP clustering,

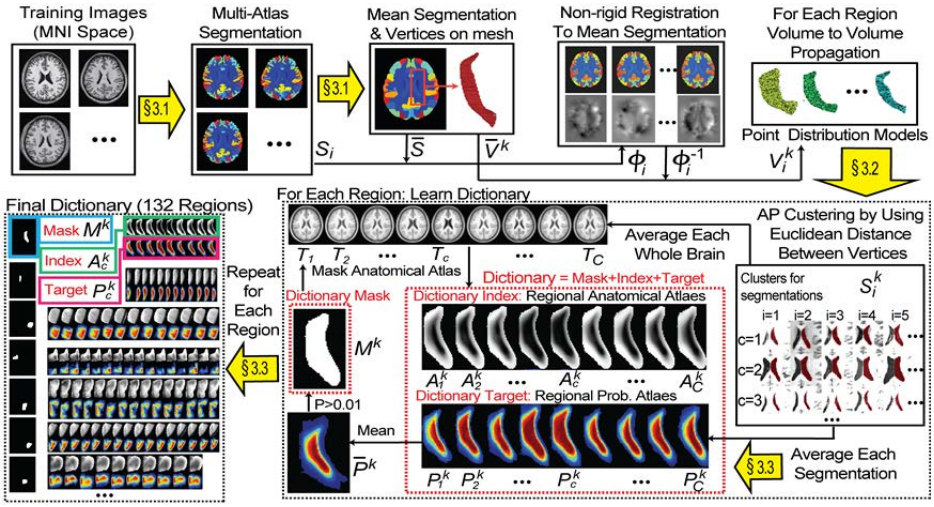

Fig. 1. Flowchart of training a data-driven dictionary of whole brain probabilistic atlas.

$$d^k(i,j) = -\frac{1}{M_k}\sum_{m=1}^{M_k}\left\|v_{i,m}^k - v_{j,m}^k\right\|^2 \quad (1)$$

where the $v_{i,m}^k$ and $v_{j,m}^k$ are the m^{th} vertex in the vertices V_i^k and V_j^k. M_k is the size of the vertices V_i^k or V_j^k. Typically, 7~20 reliable clusters are generated for each region.

3.3 Learn Dictionary

For One Region

The regional anatomical atlases A_c^k are the "dictionary index" and the regional probabilistic atlases P_c^k corresponding "dictionary target" (red rectangular in **Figure 1**). First, the regional probabilistic atlases P_c^k for the cluster c is obtained by averaging the segmentations that belong to that cluster.

$$P_c^k = \frac{1}{L_c}\sum S_i^k, \quad T_c = \frac{1}{L_c}\sum I_i, \quad all\ i \in cluster\ c \quad (2)$$

where S_i^k is the segmentation of region k from subject i and L_c is the number of segmentations in the cluster c. The anatomical atlases for each cluster are found by (2) and I_i is the whole brain anatomical image from subject i.

However, as shown in **Figure 1**, each T_c is a whole brain anatomical atlas rather than a regional anatomical atlas for region k. So, we need to extract the target area for region k by a reasonable mask M^k.

To get the mask M^k, we (1) average all $\{P_c^k\}_{c=1,2,\ldots,C}$ to $\bar{P}^k$ (2) obtain the M^k by setting the threshold $\bar{P}^k > 0.01$. The obtained mask will be much larger than any individual segmentation, which covers the potential spatial locations of region k.

Finally, we apply the mask M^k on every T_c to get a regional anatomical atlas A_c^k

$$A_c^k = T_c \circ M^k \quad (3)$$

The masked A_c^k is corresponding to the regional probabilistic atlas P_c^k.

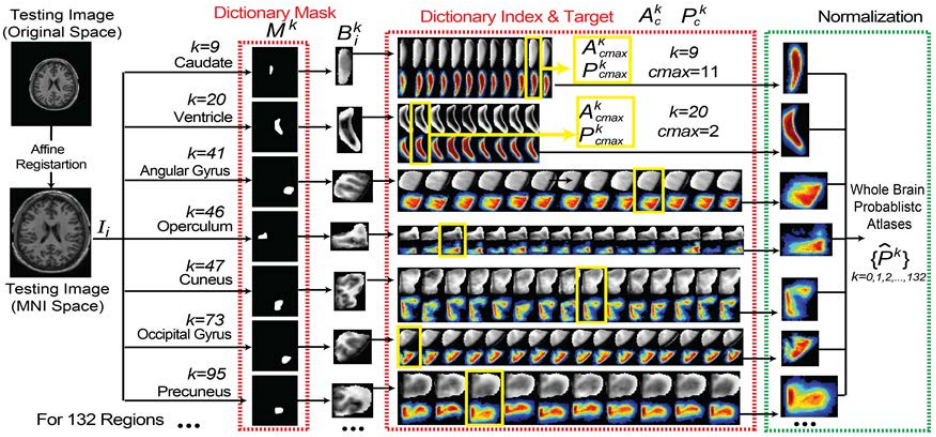

Fig. 2. Flowchart of applying the dictionary to customize a probabilistic atlas for a new subject.

For Whole Brain
We repeat the "For One Region" steps 132 times (for all regions except background) to get the whole brain dictionary as shown in the lower left part of **Figure 2**.

3.4 Apply Dictionary on New Subjects

To efficiently establish an individual whole brain probabilistic atlases, each target subject is affinely aligned [3] to the MNI305 atlas to get I_i (**Figure 2**). Then, the regional intensity B_i^k can be masked out by

$$B_i^k = I_i \circ M^k \tag{4}$$

By comparing the B_i^k to our learned dictionary, the index can be obtained by finding the most correlated regional anatomical atlas A_c^k. The correlation metrics used here is the Pearson correlation. Once the index c_{max} is found, the corresponding $P_{c_{max}}^k$ is chosen as the regional probabilistic atlas for the new subject.

$$c_{max}^k = \arg\max_c \ corr\left(A_c^k, B_i^k\right), \qquad c \in \{1,2,\dots,C\} \tag{5}$$

Repeating equations (4) and (5) for all regions, we find the 132 most correlated regional probabilistic atlases for the new subject.

3.5 Normalize to Whole Brain Atlas

Since the regional probabilistic atlases were chosen independently, the total probability for a voxel might be larger or smaller than 1. To normalize them to a complete set of whole brain probabilistic atlases, we employed a whole brain tissue probabilistic mask $M^t$from 1888 training image which contains the voxels with tissue probability greater than 0.95. For each voxel (x, y, z) within the mask M^t, the 132 regional probabilistic atlases are normalized to 1; otherwise we keep it untouched.

$$\hat{P}^k(x,y,z) = \begin{cases} \dfrac{P_{c_{max}^k}^k(x,y,z)}{Z} & x,y,z \in brain\ mask\ M^t, or\ Z > 1 \\ P_{c_{max}^k}^k(x,y,z) & otherwise \end{cases} \tag{6}$$

$Z = \sum_{k=1}^{132} P_{c_{max}^k}^k(x,y,z)$ is the normalization term.

Last, the probability of background $\hat{P}^0(x,y,z)$ is obtained by

$$\hat{P}^0(x,y,z) = 1 - \sum_{k=1}^{132} \hat{P}^k(x,y,z) \tag{7}$$

The set of $\{\hat{P}^k(x,y,z)\}_{k=0,1,2,\dots,132}$ is the normalized data-driven whole brain probabilistic atlases for the new subject. For each voxel in the whole brain probabilistic atlases, the total probability of 132 labels and background is 1.

4 Experimental Results

Two metrics are employed in the experiments. First, the Jensen-Shannon (JS) divergence is used to assess the spatial similarity between the probabilistic atlases and the

target segmentations for each testing subject [11]. Here, the "target segmentations" means the multi-atlas segregations for the withheld testing images and the manual segmentations for the OASIS images. The smaller JS divergence value is, the more similar the two spatial distributions are. So, smaller is the better for JS.

Second, to compare the different probabilistic atlases more intuitively, we apply "naive segmentation" on whole brain by choosing labels with the highest probability for each voxel. Notice that we are not providing a novel segmentation algorithm. Instead, we compare the spatial accuracy of different probabilistic atlases by using the naïve segmentation since this approach is entirely depending on the probability. Then, the Dice similarity measures the overlaps between the naive segmentations and the target segmentations.

All statistical significance tests are made using a Wilcoxon signed rank test ($p<0.01$). Creating a whole brain probabilistic atlas for a new subject can be done with 1 rigid registration and 12 seconds of CPU time (Xeon W3520 2.67GHz).

4.1 Evaluation by Withheld Testing Data

Figures 3 and **4** show the results using withheld testing subjects. The green boxplots represent the average JS or Dice values by applying the probabilistic atlases from all the other 17 group atlases for one testing subject. The blue, red and orange boxplots

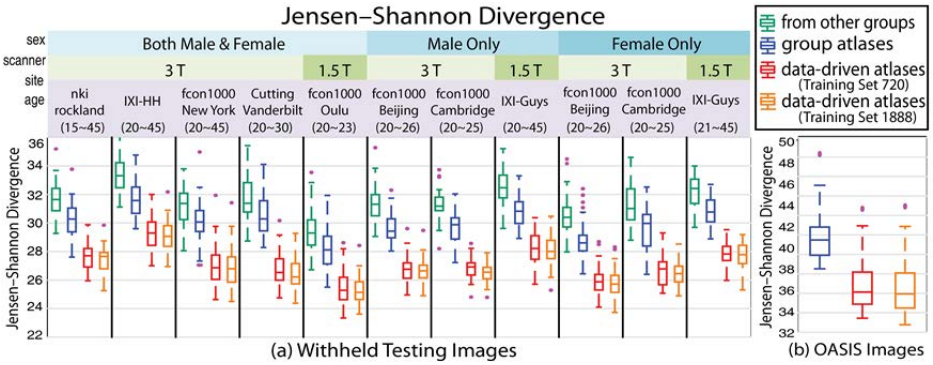

Fig. 3. Jensen-Shannon divergence. The comparisons of JS divergence for different atlases are all significantly different for both withheld and OASIS testing images.

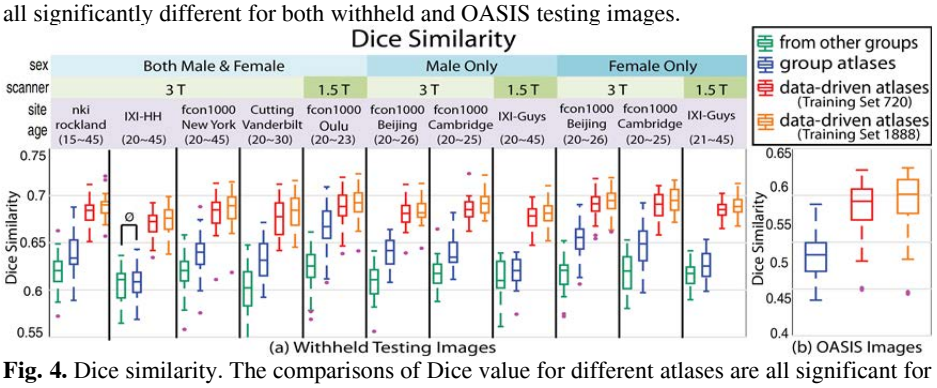

Fig. 4. Dice similarity. The comparisons of Dice value for different atlases are all significant for both withheld and OASIS testing images except the IXI-HH group marked by "Ø".

show the JS or Dice values by using the corresponding group probabilistic atlases, data-driven probabilistic atlases from Training Set 720 and from Training Set 1888.

Figure 3 and **4** demonstrate that the data-driven atlases match the target segmentations significantly better than the traditional group based atlases with the significantly smallest JS divergence and greatest Dice values while the atlases from other groups perform the worst. Moreover**,** for the data-driven atlases with two different numbers of training images, the large-scale Training Set 1888 performs significant better than Training Set 720 for both JS divergence and Dice similarities.

To conclude, (1) the group based atlases perform significantly better than the atlases from other groups which demonstrates the group-based framework is able to control the inter-group variability; (2) our proposed data-driven framework produced the more accurate probabilistic atlases than group based atlases by capturing the individual variance; (3) by using the large-scale training data, the performance of data-driven framework is improved significantly.

4.2 Evaluation by OASIS Data

45 subjects from OASIS dataset with manual segmentations are used for 44 leave one tests. The data-driven probabilistic atlases are obtained from the learned dictionary. The right hand panel of results in **Figures 3 and 4** show that the results of manual segmentations repeat the finding in §4.1.

Moreover, we show one testing subject (slice z = 75 in MNI space from 3D image) from the OASIS dataset in **Figure 5**. By comparing with the manual segmentations for 6 regions, it shows that the data-driven atlases match the true segmentations more accurately than the group atlases. Moreover, the large-scale Training Set 1888 matches the manual segmentation better than the smaller Training Set 720.

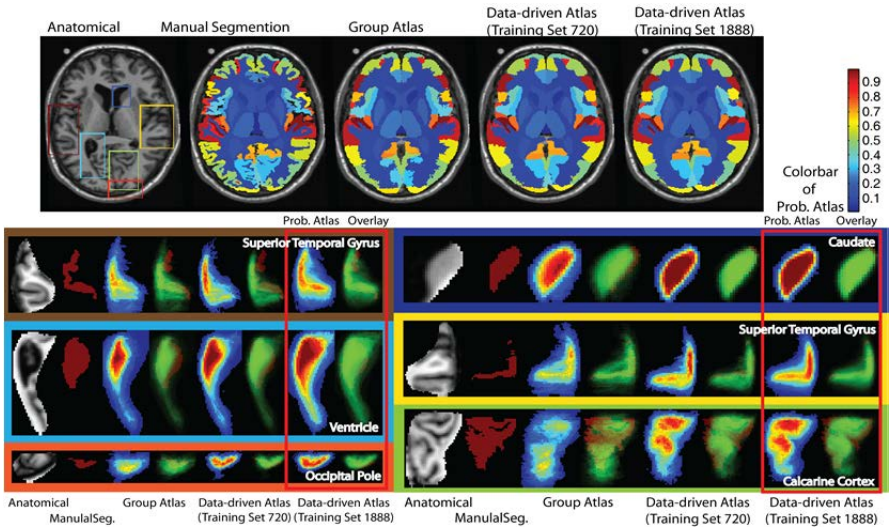

Fig. 5. One testing subject from OASIS dataset. Top row shows the anatomical image, manual segmentation, highest probability segmentations using the group probabilistic atlases, Training Set 720 and Training Set 1888. The lower rows show the details of 6 regions. For each region, from left to right are: anatomical image, manual segmentation, probabilistic atlases generated by different methods and their overlays on manual segmentations.

5 Discussion

We present a data-driven framework to learn a dictionary of whole brain probabilistic atlases to achieve accurate individualized whole brain probabilistic atlases. This framework (1) provides a new perspective of using data-driven scheme rather than the traditional group based methods, (2) uses the large-scale heterogeneous data to achieve more personal specific probabilistic atlases than using the single-group and single-site data by capturing the individual variation (3) demonstrates the advantages of using large-scale scheme in generating personal probabilistic atlases compared with the smaller size of training images, and (4) only requires one affine registration and Pearson correlations to apply to new subjects which achieves low computational cost.

Due to the higher accuracy and low computational cost, the proposed method is able to be the priors in many medical image processing algorithms and applications.

Acknowledgments: This research was supported by NIH 5R21EY024036, NIH 1R21NS064534, NIH 2R01EB006136, NIH 1R03EB012461, NIH R01EB006193 and also supported by the Intramural Research Program, National Institute on Aging, NIH.

References

1. Shattuck, D.W., et al.: Construction of a 3D probabilistic atlas of human cortical structures. NeuroImage 39, 1064-1080 (2008)
2. Commowick, O., et al.: Using Frankenstein's creature paradigm to build a patient specific atlas. Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention 12, 993-1000 (2009)
3. Ourselin, S., et al.: Reconstructing a 3D structure from serial histological sections. Image Vision Comput 19, 25-31 (2001)
4. Evans, A.C., et al.: 3D statistical neuroanatomical models from 305 MRI volumes. In: Nuclear Science Symposium and Medical Imaging Conference, 1993., 1993 IEEE Conference Record., pp. 1813-1817. IEEE (1993)
5. Avants, B.B., et al.: Symmetric diffeomorphic image registration with cross-correlation: evaluating automated labeling of elderly and neurodegenerative brain. Medical image analysis 12, 26-41 (2008)
6. Asman, A.J., Landman, B.A.: Non-local statistical label fusion for multi-atlas segmentation. Medical image analysis 17, 194-208 (2013)
7. Wang, H., et al.: A learning-based wrapper method to correct systematic errors in automatic image segmentation: consistently improved performance in hippocampus, cortex and brain segmentation. NeuroImage 55, 968-985 (2011)
8. Klein, A., et al.: Open labels: online feedback for a public resource of manually labeled brain images. In: 16th Annual Meeting for the Organization of Human Brain Mapping. (2010)
9. Heimann, T., Meinzer, H.P.: Statistical shape models for 3D medical image segmentation: a review. Medical image analysis 13, 543-563 (2009)
10. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. Science 315, 972-976 (2007)
11. Gouttard, S., et al.: Assessment of reliability of multi-site neuroimaging via traveling phantom study. Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention 11, 263-270 (2008)

A neuroscience gateway for handling and processing population imaging studies

M.W.A. Caan, J. Teeuw, S. Shahand, M. M. Jaghoori, J. Huguet, A. van Altena, and S.D. Olabarriaga

Academic Medical Center, Amsterdam, Netherlands
m.w.a.caan@amc.nl

Abstract. Data handling and processing in population imaging studies on high-end resources requires dedicated technological knowledge, something for which neuroscientists are commonly not trained. We have developed a NeuroScience Gateway that provides an intuitive web-based interface for data-intensive science discovery and hides the underlying technological details from the scientist. The eXtensible Neuroimaging Archive Toolkit (XNAT) was adopted for storing medical imaging data and processing results. The e-BioInfra Catalogue holds a database of metadata, user, and system information. The processing manager performs and monitors data processing on a grid infrastructure. Processed data are archived and a data history report is generated for provenance purposes. Over a period of 18 months, 36 users have processed their data, amounting to 6.85 CPU years. We review the lessons learned over this period of time and sketch an outlook for future development.

1 Introduction

Due to the sheer amount of produced data, population imaging studies require high-end storage and processing resources to answer research questions on, e.g., ageing processes and neurodegenerative disorders. Distributed computing infrastructures, such as grids, enable processing of large datasets, but dedicated technical knowledge is needed to exploit them. Most neuroscientists and bioscientists are not trained to use interfaces that expose low-level details of the technology. To bridge this gap, Science Gateways (SGs) have been proposed as easy-to-use (high-level) user interfaces to enable data-intensive scientific discovery without knowing the underlying technical details.

In recent years, different projects have been started to develop SGs in the neuroscience field. Examples include the data engine [6] of the CHAIN project [1], COINS, a neuroimaging tool suite [11], a web-based, distributed computing platform coined CBRAIN [15], and the LONI pipeline [5], a graphical workflow environment. The neuGRID for you (N4U) SG [2] houses multiple algorithms, pipelines, and visualization toolkits on grid, cloud, and clusters. The eXtensible Neuroimaging Archive Toolkit (XNAT) is developed with data archiving and sharing as its main goal, but also supports running pipelines per individual

M.W.A. Caan, J. Teeuw, S. Shahand, M. M. Jaghoori, J. Huguet, A. van Altena, S.D. Olabarriaga ; A neuroscience gateway for handling and processing population imaging studies, In: *Proceedings of the 1st Miccai 2015 Workshop on Management and Processing of images for Population Imaging* – MICCAI-MAPPING2015, C . Barillot, M. Dojat, D. Kennedy and W. Niessen (Eds), pp.15-22, 2015.

dataset [10] but not on multiple datasets simultaneously. LORIS is an alternative web-based data management system for multi-center studies [4].

A population imaging study may be described by a number of research study phases: study design, data acquisition, data handling, processing, analysis, and publication [12]. A SG for computational neuroscience should support all of these study phases. The Neuroscience gateway (NSG) presented in this paper [13] focuses on the data- and compute-intensive elements, i.e., the data handling and processing phases. It should therefore support metadata collection, enable data processing and provenance management, provide sufficient security and privacy mechanisms, enable data and methodology sharing, and offer scalable, transparent, and flexible management of storage and computing resources. In this paper, we present an overview of the NSG system's architecture, usage statistics and lessons learned over the past years of NSG activity.

2 Overview of NSG

As the NSG is presented in details in [13], here we highlight the most relevant aspects to support the discussions.

Usage scenario. Users log into the system via an intuitive web interface to access the imaging studies of which they are a member to browse, search, and filter data and metadata. They select the subjects and their corresponding MR imaging scans to be processed by one of the available applications, and then launch the processing. The NSG manages data transport from the data server to the grid, then schedules and monitors the data processing applications on the grid. The resulting output is stored back on the data server with relevant provenance information for future reference. The gateway also handles authentication with the data server and DCI resources transparently.

System architecture. Figure 1 shows the main system components. The MRI-scanner is inside a hospital. The acquired data are pseudonymized and sent from the scanner to the internal data server (XNAT) located behind the hospital's firewall. From here, the data are anonimized more strictly and uploaded to an external data server (eXNAT) accessible via the internet. The gateway is hosted on a server in the demilitarized zone (DMZ) of the hospital network, a semi-secured layer between the heavily protected intranet and the (open) internet.

Data server. XNAT was adopted for storing medical imaging data and the corresponding metadata. XNAT [10] is an open-source information management system that offers an integrated framework for data storage and management. It provides a RESTful API with a set of web services that enable querying for data and metadata, uploading, modifying and downloading resources. The API eases the programmatic integration of additional tools and systems with XNAT. Its data model provides a general hierarchy and security model: data is grouped under logical 'Projects' containers which are stored as metadata. A project contains data of a single imaging study, accessible to project members only. Each project contains a set of subjects, which can be measured in events abstractly

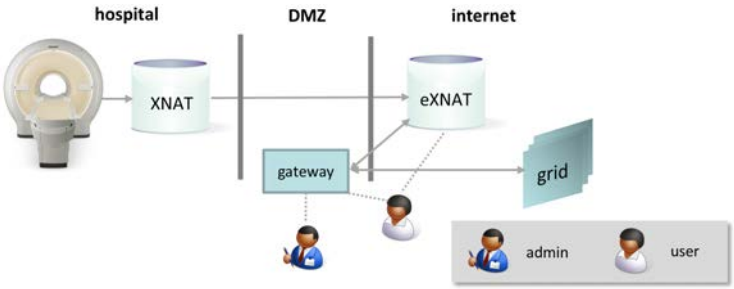

Fig. 1: System architecture in which the NSG is embedded.

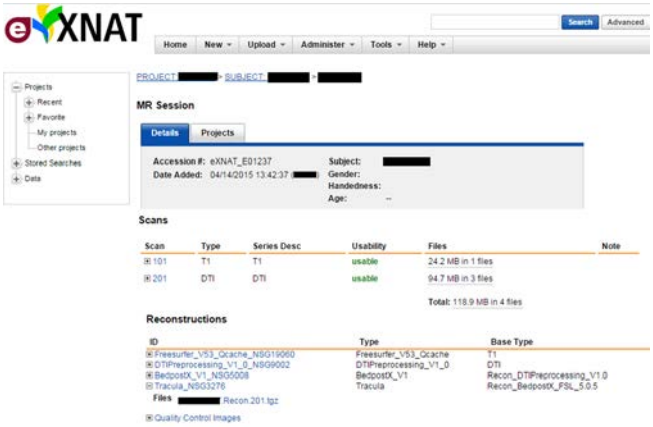

Fig. 2: Screenshot of the external XNAT server showing reconstructions processed on the NSG for one subject at the bottom of the screen.

referred to as 'Experiments'. Image sessions are specific representations of experiments where imaging data is acquired. Externally processed data can be stored as a 'Reconstruction'. See a screenshot of the eXNAT web interface in Figure 2.

Gateway components. Data and metadata management functionalities are implemented in the e-BioInfra Catalogue (eCAT). User and system-level information is stored using a model with the following basic entities: User, Project, Data, Metadata, Resource, Credential, Application, Processing, Submission, and Submission Status. Data transport from the XNAT to the grid is managed by a separate component, the processing manager (PM) [8], which interacts with the eCAT to obtain the relevant information and credentials. The PM prepares, submits, and monitors jobs with selected applications and data. The current NSG applications are defined as workflows with WS-PGRADE/gUSE [9]. If the PM detects failures in processing the grid jobs, the administrator is notified,

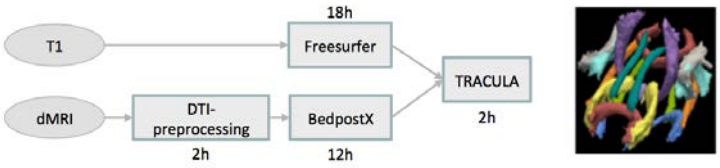

Fig. 3: Pipeline to run TRACULA application including preprocessing DTI data, image segmentation (Freesurfer [7]) and estimating white matter orientation distributions (FSL BedpostX [16]), with average grid execution times.

who will take the required actions (e.g., resume aborted experiments and notify the user). The NSG provides the user with monitoring information (figure 4).

Security. Users are provided with username/password to gain access to the gateway and to XNAT. These are kept as two separate credentials to preserve full control for the data server administrator. The NSG manages multiple credentials transparently. A robot X-509 certificate is used to access grid resources, which is managed by the gateway administrator transparently for the users.

Example of application. TRACULA is a method available at the NSG for automatically reconstructing white matter pathways from T1- and diffusion-weighted MRI data [17] - figure 3 displays the used pipeline. TRACULA takes as input two MRI scans per scanning session of a subject: a T1-weighted structural scan and a diffusion weighted MRI scan. The output is a series of reconstructed white matter tracts and their statistics. In the NSG web interface the user selects the data and runs the applications, one at a time. The UI guides the user through the pipeline steps by recommending suitable applications based on selected data. Figure 2 illustrates how the results of the various steps are stored by the NSG as Reconstructions with metadata in XNAT.

3 Results

The NSG has been used by neuroscientists since the end of 2013 for analysis of large imaging studies. The data handling and processing phases in research are supported as follows. After logging into the NSG, the user is provided with a list of projects on the eXNAT server, for which access has been granted. The user is then directed to the Data tab, to select scans to be processed. Figure 4 illustrates the interface for monitoring on-going or completed processing tasks. For each processed scan, a data history report is generated as illustrated in Figure 5. The report contains the main input data characteristics, such that it can be traced back in XNAT; the selected application and its version number; the experiment date; the generated files; and the involved researcher.

Usage statistics were collected over a period of 18 months. In total 40 users registered to the NSG, of which 36 launched at least one experiment. Figure 6 provides an overview of the number of jobs per user, and the estimated CPU-time used. The graphs reflects the difference in size of datasets processed by the

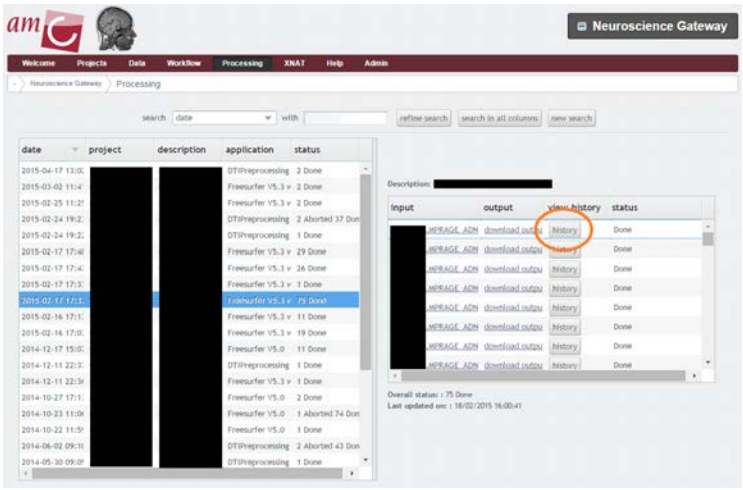

Fig. 4: Screenshot os NSG monitoring window with a list of experiments (left) and the included datasets (right). The History button (circle) provides provenance information.

Table 1: Total grid usage by the NSG: number of jobs and estimated CPU-years.

	Total	Freesurfer	DTIpreprocessing	BedpostX	TRACULA
#jobs	6325	2527	1426	696	1676
CPU-years	6.85	5.19	0.33	0.95	0.38

users, and teh ability of the NSG to scale well with these datasets. The Freesurfer application is using most computing resources. Processing one dataset using Freesurfer on average amounted to 18 hours, which is comparable to known processing times on local servers. In Table 1 the used CPU time is given, amounting to 6.85 years in total.

4 Discussion and lessons learned

The NSG presented here follows earlier SG versions that evolved based on lessons learned along time [3, 14]. Compared to the previous version [14], the data flow currently follows the course of a scientific experiment more naturally. Previously, the user selected the desired application and then needed to manually upload prepared archives containing the data to be processed. This process was error prone because the archives needed to be formatted in a strict manner. Also, no metadata was stored with the processed results, which made management and reuse more difficult in the long term. These difficulties have been removed by the adoption of a third-party data server, XNAT, which is widely used by the neuroimaging community. Nevertheless, additional lessons have been learned in

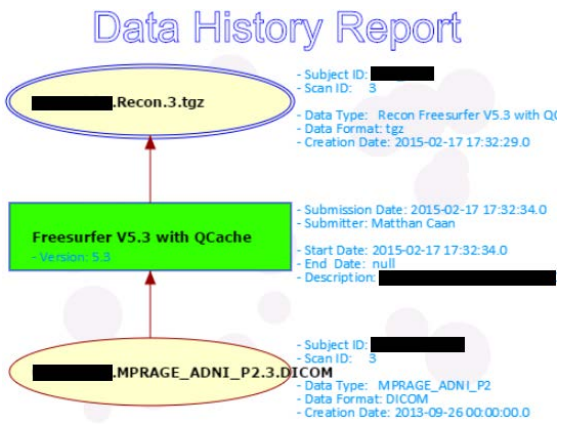

Fig. 5: Data history report captured and gathered automatically for a data processing.

the past year from how users in fact work with the NSG, and how well this matches to the scenario we had initially designed.

A major difficulty derived from our choice of integrating XNAT with the NSG, while keeping them as independent systems. The user is currently confronted with two credentials, and, although the XNAT interface is embedded in the NSG, most users interact with the two systems separately. It also requires training for users to conceptually understand the differences between both systems. Additionally, the NSG hosts a separate metadata database, duplicating some fields from XNAT for search functionality and efficiency reasons. The two databases are synchronized periodically, which introduces latency in using NSG when new data or users are added. It also increases network traffic and load on the servers. Contrary to XNAT, the NSG currently does not offer an API.

The strategy for storing results back in XNAT is currently not optimal: the results are packed into an archive and uploaded to XNAT as a single file per processing. For Freesurfer segmentations, a XNAT Assessment data type has been developed already, ordering segmentation volumes in a hierarchical manner. This allows for querying and retrieving results in a structured manner. Moreover, since no working directory or scratch model is available for the NSG, all processed data are directly archived in XNAT, possibly leading to database pollution.

We also observed that, although XNAT provides native support to the DICOM format, it is designed in a generic way to support custom file formats. Proprietary file formats of MRI vendors form a clear example to be embedded in XNAT. We have exploited this to develop an import pipeline for the Philips PAR/REC data format, which is now actively used in our user community.

Concerning the applications, first, the current implementation does not allow running larger pipelines comprising of multiple processing steps at once. Second, multiple data inputs per application are not allowed in the web interface. In the example of TRACULA (Figure 3), one scan from the imaging session is se-

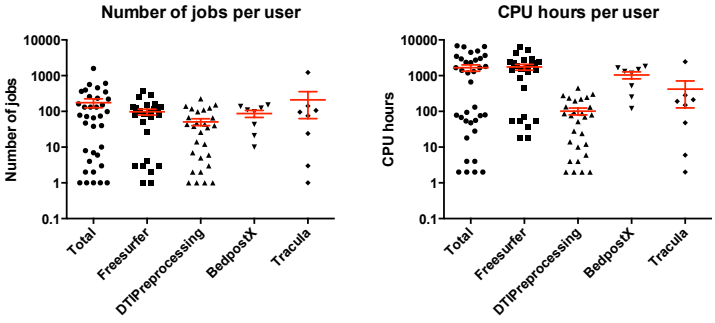

Fig. 6: Number of jobs and consumed CPU-hours per user of the NSG (log scale). Each data point corresponds to a user. A job refers to the execution of an application with a single dataset. Error bars denote the mean and standard error of the mean.

lected by the user, and the other inputs are automatically selected by the system following a series of rules specific to the application requirements. Third, all applications are predefined, because the NSG was conceived and designed to target neuroscience users with little to none scripting and programming background. Future work should provide a solution for user-defined imaging pipelines and changing application parameters. If needed, the gateway administrator can now create different versions of an application with specific sets of parameters for different use cases. Fourth, the NSG only allows for fully automated processing of data, without possibilities for semi-automated or interactive processing (e.g. in image segmentation). Finally, one important aspect for third-party applications is to properly arrange licensing. Freesurfer is for example issued under a public license, valid for single users only. Therefore all NSG users are requested to apply for a Freesurfer license individually before being allowed to run the application.

Despite the above mentioned limitations, a group of 36 users has been able to successfully adopt to process data of large studies using the NSG. As illustrated in Figure 6, on average each user processed more than 100 jobs, and Freesurfer and DTI Preprocessing are the most popular. The popularity of Freesurfer and related CPU-hours consumed confirms high demands for greater processing capacity. This is a significant step forward, compared to the practice of manual, slow and error-prone processing of data. This opens the road for adopting the NSG in larger population imaging studies that lie ahead, for which we need to make our service and code publicly available.

Acknowledgements. We thank colleagues who participated in NSG development along time, in particular Juan Luis Font. The NSG uses resources from the Dutch e-Science Grid with the support of SURF Foundation. The NSG is funded by various projects and organizations: COMMIT/, HPCN Fund of the University of Amsterdam, SCI-BUS and ER-flow FP7 projects.

References

1. Co-ordination and Harmonisation of Advanced e-INfrastrucures for Research and Education Data Sharing. www.chain-project.eu
2. The N4U (neuGRID for you) Project website. http://neugrid4you.eu
3. Caan, M., Shahand, S., Vos, F., van Kampen, A., Olabarriaga, S.: Evolution of grid-based services for diffusion tensor image analysis. Future Generation Computer Systems 28(8), 1194 – 1204 (2012)
4. Das, S., Zijdenbos, A.P., Harlap, J., Vins, D., Evans, A.C.: LORIS: a web-based data management system for multi-center studies. Frontiers in neuroinformatics 5, 37 (2011)
5. Dinov, I., Lozev, K., Petrosyan, P., Liu, Z., Eggert, P., Pierce, J., Zamanyan, A., Chakrapani, S., Van Horn, J., Parker, D.S., Magsipoc, R., Leung, K., Gutman, B., Woods, R., Toga, A.: Neuroimaging study designs, computational analyses and data provenance using the LONI pipeline. PloS one 5(9), e13070 (2010)
6. Fargetta, M., Rotondo, R., Barbera, R.: A data engine for grid science gateways enabling easy transfer and data sharing. Presentation in the EGI community Forum 2012 (March 2012)
7. Fischl, B.: FreeSurfer. NeuroImage 62(2), 774–781 (Aug 2012)
8. Jaghoori, M.M., Shahand, S., Olabarriaga, S.D.: Processing manager for science gateways. In: IEEE 7th International Workshop on Science Gateways (2015)
9. Kacsuk, P., Farkas, Z., Kozlovszky, M., Hermann, G., Balasko, A., Karoczkai, K., Marton, I.: WS-PGRADE/gUSE generic DCI gateway framework for a large variety of user communities. Journal of Grid Computing 10(4), 601–630 (2012)
10. Marcus, D.S., Olsen, T.R., Ramaratnam, M., Buckner, R.L.: The Extensible Neuroimaging Archive Toolkit: an informatics platform for managing, exploring, and sharing neuroimaging data. Neuroinformatics 5(1), 11–34 (2007)
11. Scott, A., Courtney, W., Wood, D., de la Garza, R., Lane, S., King, M., Wang, R., Roberts, J., Turner, J.A., Calhoun, V.D.: COINS: An Innovative Informatics and Neuroimaging Tool Suite Built for Large Heterogeneous Datasets. Frontiers in neuroinformatics 5, 33 (2011)
12. Shahand, S., Caan, M., Van Kampen, A., Olabarriaga, S.: Integrated support for neuroscience research: From study design to publication. In: Studies in Health Technology and Informatics. vol. 175, pp. 195–204 (2012)
13. Shahand, S., Benabdelkader, A., Jaghoori, M.M., al Mourabit, M., Huguet, J., Caan, M.W.A., van Kampen, A.H.C., Olabarriaga, S.D.: A Data-Centric Neuroscience Gateway: Design, Implementation, and Experiences. Concurrency and Computation: Practice and Experience 27(2), 489–506 (2015)
14. Shahand, S., Santcroos, M., Kampen, A., Olabarriaga, S.: A grid-enabled gateway for biomedical data analysis. Journal of Grid Computing 10(4), 725–742 (2012)
15. Sherif, T., Rioux, P., Rousseau, M.E., Kassis, N., Beck, N., Adalat, R., Das, S., Glatard, T., Evans, A.C.: CBRAIN: a web-based, distributed computing platform for collaborative neuroimaging research. Frontiers in neuroinformatics 8, 54 (2014)
16. Woolrich, M.W., Jbabdi, S., Patenaude, B., Chappell, M., Makni, S., Behrens, T., Beckmann, C., Jenkinson, M., Smith, S.M.: Bayesian analysis of neuroimaging data in FSL. NeuroImage 45(1), S173–S186 (Mar 2009)
17. Yendiki, A., Panneck, P., Srinivasan, P., Stevens, A., Zöllei, L., Augustinack, J., Wang, R., Salat, D., Ehrlich, S., Behrens, T., Jbabdi, S., Gollub, R., Fischl, B.: Automated probabilistic reconstruction of white-matter pathways in health and disease using an atlas of the underlying anatomy. Frontiers in Neuroinformatics 5, 23 (2011)

Shanoir: Software as a Service Environment to Manage Population Imaging Research Repositories

Christian Barillot[1,2,3,*], Elise Bannier[1-4], Olivier Commowick[1,2,3], Isabelle Corouge[1,2,3], Justine Guillaumont[1,2,3], Yao Yao[1,2,3] , Michael Kain[1,2,3]

1 Inria, VisAGeS Project-Team, F-35042 Rennes, France
2 INSERM, U746, F-35042 Rennes, France
3 University of Rennes I, CNRS, UMR 6074, IRISA, F-35042 Rennes, France
4 CHU Rennes, Department of Neuroradiology, F-35043 Rennes, France

Abstract. Some of the major concerns of researchers and clinicians involved in population imaging experiments are on one hand, to manage the huge quantity and diversity of produced data and, on the other hand, to be able to confront their experiments and the programs they develop with peers. In this context, we introduce Shanoir, a "Software as a Service" (SaaS) environment that offers cloud services for managing the information related to population imaging data production in the context of clinical neurosciences. We show how the produced images are accessible through the Shanoir Data Management System, and we describe some of the data repositories that are hosted and managed by the Shanoir environment in different contexts.

Keywords. Population imaging, database, data sharing, neuroinformatics, "Software as a Service" (SaaS), Cloud Computing, web application, Java, web services, Shared repositories, centralized resources

1 Introduction

Some of the major concerns of researchers and clinicians involved in population imaging experiments are, on one hand, to manage the huge quantity and diversity of produced data and, on the other hand, to be able to confront their experiments and the programs they develop with peers. In practice, researchers or clinicians in the neuroimaging domain are encouraged to set up large-scale experiments, but the lack of resources and capabilities to recruit locally subjects who meet specific inclusion criteria motivates the need for sharing the load in order to produce the relevant imaging data. For these reasons, making possible the pooling of experimental results, through the Internet and between collaborative centers, allows to recruit large subject populations and to widen the scientific achievement of the conducted experimental studies. Also, through distributed imaging databases, the search for similar results, the search for images containing singularities or transverse searches via data mining techniques could highlight possible regularities. Moreover, this will broaden also the possible panel of people involved in neuroimaging studies, while protecting the excellence of the supplied work.

In this context, the Shanoir[†] (SHAring NeurOImaging Resources) environment aims at establishing the conditions allowing, through the Internet, to share distributed information sources in neuroimaging, whether these sources are located in various centers of experi-

[*] Corresponding Author: Christian.Barillot@irisa.fr / Ph : +33 299847505 / Fax : +33 299847171
[†] Shanoir: http://www.shanoir.org

Christian Barillot, Elise Bannier, Olivier Commowick, Isabelle Corouge, Justine Guillaumont, Yao Yao , Michael Kain ; Shanoir: Software as a Service Environment to Manage Population Imaging Research Repositories, In: *Proceedings of the 1st Miccai 2015 Workshop on Management and Processing of images for Population Imaging – MICCAI-MAPPING2015, C . Barillot, M. Dojat, D. Kennedy and W. Niessen (Eds), pp.23-30, 2015.*

mentation, clinical departments of neurology, or research centers in cognitive neurosciences or image processing. This enables a large variety of users to diffuse, exchange or reach neuroimaging information with appropriate access means, in order to be able to retrieve information almost as easily as if the data were stored locally by means of the "cloud computing" Storage as a Service (SaaS) concept [1].

In this paper, we introduce the Shanoir software environment that offers services for managing the information related to neuroimaging data production in the context of clinical neurosciences. We show how the produced images are accessible through the Shanoir Data Management System. The paper is organized as follows. In section 2, we rapidly describe the software environment, and their extension for loading the data for querying the data, and for processing the data. Section 3 describes some population data repositories and section 4 provides a discussion on the use of these repositories and the potential evolutions.

2 Shanoir software environment

2.1 General description of the software environment

Shanoir is an open source software environment, with QPL licensing, designed to archive, structure, manage, visualize and share neuroimaging data with an emphasis on managing distributed collaborative research projects. It provides common features of neuroimaging data management systems along with research-oriented data organization and enhanced accessibility. Shanoir is based on a secured J2EE application running on a JBoss server, reachable via graphical interfaces in a browser or by third party programs via SOAP web services. It behaves as a repository of neuroimaging files coupled with a relational database holding meta-data (**Fig. 1**).

Shanoir uses semantics for concepts organization that are defined by ontology, called OntoNeuroLOG[‡] [2] [3]. OntoNeuroLOG reuses and extends the OntoNeuroBase ontology [4]. In Shanoir, the OWL-Lite implementation was manually derived from the OntoNeuroLOG initial expressive representation to Java classes. The data model based on this ontology is devoted to the neuroimaging field and is structured around research studies whereof involved patients have examinations, which either produce image acquisitions or clinical scores. Each image acquisition is composed of datasets represented by their acquisition parameters and image files. For security and regulation reasons, by default, the system only keeps anonymous data. Raw and derived (i.e. post-processed) image files can also be imported into the system from various sources (DICOM CDs, PACS, image files in NIfTI / Analyze format) using either online wizards, with completion of related metadata, command line tools or SOAP web services. For raw data, once de-identified during import, DICOM header's content is automatically extracted and inserted into the database by a customizable feature called "Study Card".

Shanoir can also record any executed processing allowing retrieving workflows applied to a particular dataset along with the derived data. Clinical scores resulting from instrument assessments (e.g. neuropsychological tests) can be recorded and easily retrieved and exported in different formats (Excel, CSV, XML). Scores, image acquisitions and post-processed images are bound together, which makes relationship analysis possible. The instrument database is scalable and new measures can be added in order to meet specific project needs.

Using cross-data navigation and advanced search criteria, the user can quickly point to a subset of data to be downloaded. Client side applications have also been developed to illus-

[‡] OntoNeuroLOG: http://neurolog.i3s.unice.fr/public_namespace/ontology

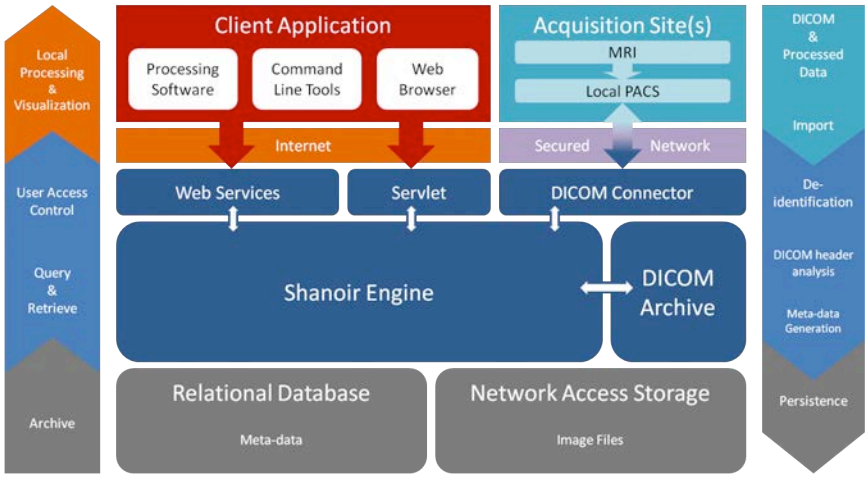

Fig. 1. Shanoir Software Architecture

trate how to locally access and exploit data though the available web services. With regards to security, the system requires authentication and user rights are tunable for each hosted studies. A study manager can thereby define the users allowed to see, download or import data into his/her study or simply make it public.

In practice, Shanoir serves neuroimaging researchers in organizing efficiently their studies while cooperating with other laboratories. By managing patient privacy, Shanoir allows the exploitation of clinical data in a research context. It is finally a handy solution to publish and share data with a broader community.

2.2 The Study Card and quality control concepts

Images can be imported in Shanoir from various sources: DICOM CDs, PACS (with DICOM Query & Retrieve), and image files (in NIfTI and Analyze format). Users are guided step by step through online forms to perform imports. In addition of archiving DICOM files, NIfTI copies are automatically generated and saved. This is convenient since the NIfTI format is better suited to local 3D image processing (such as registration, segmentation, statistical analysis, etc.) than the DICOM format.

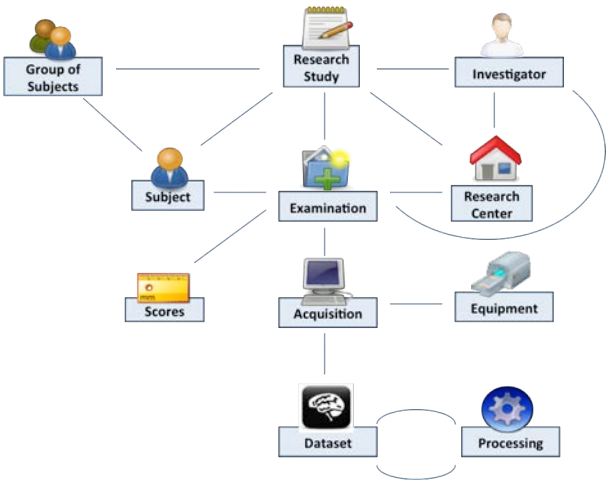

Fig. 2. Shanoir data organization

The Study Card concept

While being archived, the DICOM files are processed in two phases. The first phase de-identifies the images. The second one inserts into the database the metadata items generated from the DICOM header. This is achieved thanks to the "Study Card" concept. This concept allows the online meta-data wrapping between the local data to be imported (center, acquisition equipment…) and the semantic concepts of the research study the data will be assigned to. This allows the alignment between the actual DICOM metadata to the ontology, and also provides additional allocation of concepts to the stored images that is more related to the research study protocol (e.g. functional MRI, perfusion imaging, contrast agent, diffusion imaging…). The mechanism behind this feature is based on a user-predefined set of rules associating to particular acquisition equipment, and a particular data production site to the desired research study. Each rule determines the specific value of a metadata item according to the value(s) of one or more specific DICOM tag(s) (e.g. Series Description…). This greatly facilitates the consistent recording and alignment to the ontology of metadata for all the data of a research study without tedious workflow during the online import of images. Due to the simplicity of the process, no specific skills are prerequired to perform the import of data and it only takes a few minutes over the Internet. This "Study Card" concept also allows an automatic quality control of the data imported based on their metadata. For instance, a conformal statement can be attached to the imported data according to a match score to the Study Card rules.

2.3 The web portal

Shanoir provides a user-friendly secure web access and offers an intuitive workflow to ease the collection and retrieval of neuroimaging data from multiple sources (Fig. 3). On the home page, the user can access to the most frequent functionalities: Find and Download Datasets, Explore the Research Studies, Find Clinical Scores, and Import Data. On the top of all pages, the user always has a very complete navigation menu that leads to all services.

2.4 The interoperability

Interoperability is a very important concern for the Shanoir environment. For this purpose, Shanoir offers a web service interface that is open to all possible clients. This interface is already used by different external applications, developed either in C++, Java or Objective-C environments such as ShanoirUploader, medInria (http://med.inria.fr) and Shanoir.

SOAP for the integration of services.

Shanoir web services interfaces are based on the Simple Object Access Protocol (SOAP). Messages between client and server that are exchanged based on XML, with defined elements. As transport layer HTTP on base of TLS is used. The elements and services are described with the Web Service Description Language (WSDL). On base of this description client stubs can be automatically generated to simplify the connection of new clients. The web service layer is implemented with the Java API for XML web services.

"ShanoirUploader" for seamless integration of data.

"ShanoirUploader" is a Java desktop application that transfers data securely between a PACS and a Shanoir server instance (e.g. within a hospital). It uses a DICOM query/retrieve connection to search and download images from a local PACS. After retrieval, the DICOM files are locally anonymized (using either a built-in process or a custom one)

and then uploaded to the Shanoir server. The primary goals of that application are to enable mass data transfers between different remote server instances and therefore reduce the waiting time of the users, when importing data into Shanoir. Most of the time during import is spent with data transfers.

Apache Solr for metadata querying.

Shanoir integrates the enterprise search platform, Apache Solr (http://lucene.apache.org/Solr/), to provide the users a vast array of advanced features such as near real-time indexing and queries, full-text search, faceted navigation, autosuggestion and autocomplete. One of the most important features of Solr search is the faceted navigation. Facets correspond to properties of the Solr information elements. They are derived by analysis of the pre-existing meta-data that are related to the ontology model used by Shanoir. All the metadata are indexed in a JBoss server that hosts the Solr servlets. A custom security post-filter has been also developed and implemented in Shanoir for user access control. This filter retrieves user identification and access rights in Shanoir and interacts with the Solr server to show pertinent results that the user is allowed to access.

MedInria for image processing.

Shanoir web services may also be queried from standalone C++/Qt applications through the QtShanoir library (http://qtshanoir.gforge.inria.fr). QtShanoir uses the SOAP-based web services provided by a Shanoir server to get and display studies, patients, and data with their associated metadata. In QtShanoir, a set of Qt widgets are defined that can be embedded in any Qt application. This library was used to implement a Shanoir query plugin inside the medInria visualization and processing software. This implementation allows for the interrogation and the download of image data from Shanoir, to process it within medInria using the available processing tools and then upload back the processing results to the Shanoir server with the correct metadata values (**Erreur ! Source du renvoi introuvable.**).

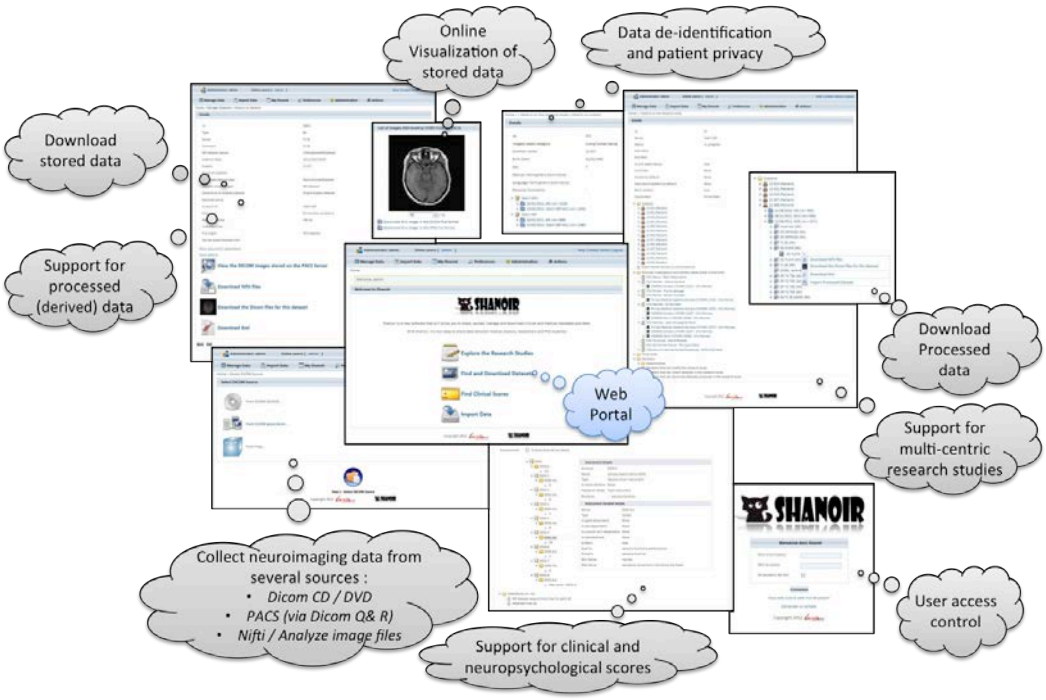

Fig. 3: Shanoir web portal summary of the main functionalities

3 Data repositories

Each Shanoir repository has an administrator that manages the access rights of the repository. Then, each user has to request an account through the dedicated web form and specify which study he/she wants to access, who is his/her contact, what will be his/her role concerning this study, and what level of expertise/access is needed (guest, user, expert, admin)... According to the information provided, the Shanoir administrator of the repository grants (or not) the user access to the system. The access to a specific study is granted by the person responsible of this study (i.e. the PI of the research study or its official representative). Depending on these settings, the new user will be able to see, download, and import datasets or even to modify the study parameters. The corresponding rights are set for a limited time and must go through a renewal process on purpose. If requested, the user can receive a report by email each time data are imported in his/her study.

3.1 The Shanoir@Neurinfo Repository

Started in 2009, the Neurinfo research facility[§] promotes translational clinical research and supports the development of clinical research, technological and methodological activities. It offers resources for in vivo human imaging acquisition, image data analysis and image data management. A large community of users, both clinicians and scientists, uses these resources as part of local, national or international imaging based research projects.

All the data produced at Neurinfo for academic or clinical research purposes are managed through a dedicated Shanoir@Neurinfo repository (**Fig. 4**) administered by the facility technical staff. The Shanoir@Neurinfo server also hosts data from multi-sites imaging studies. In total, more than 1To data from 31 centers and 37 scanners are archived within this repository (see Left Table in **Fig. 4** for details).

On the daily practice, DICOM data are imported by a technician from either a local PACS, a CD/DVD or a disk drive containing the DICOMDIR at its root and the DICOM files. The clinical studies conducted at Neurinfo concern the whole-body (brain, spine, heart, lung, pelvis, vasculature...) with a major focus on brain anatomy and function in normal control and pathological populations. Out of the 60 or so ongoing research studies at the Neurinfo platform, 75% relates to brain imaging, 15% to abdominal imaging and

CENTERS	**31**
ACQUISITION EQUIPMENTS (MRI SCANNERS)	**37**
RESEARCH STUDIES	**60**
SUBJECTS	**2.228**
EXAMINATIONS	**3.157**
DATASET ACQUISITIONS	71.993
DATASETS	114.441
ACTIVE USERS	**52**
TOTAL USERS	*127*
PACS Shanoir (DICOM)	1588 Go
Nifti Data	1511 Go
Data Base	5 Go

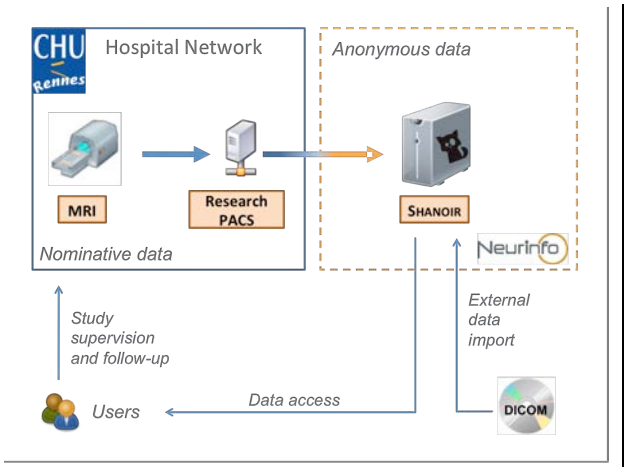

Fig. 4. The Shanoir@Neurinfo repository: Current Global Statistics (*left*) and Service Infrastructure (*right*)

[§] http://www.neurinfo.org/

10% to heart imaging. Among the neuro-imaging clinical studies, multiple sclerosis, dementia, tumors, stroke and mood disorders are the most investigated pathologies.

The general policy for the Shanoir@Neurinfo repository for dissemination of the data related to one study is decided upon beforehand with the principal investigator, in agreement with the informed consent form approved by the ethics committee and signed by the participant. Any opening of the data to third parties is submitted to the approval of the principal investigator prior to allow the (complete or partial) access to a third party user. Nonetheless, to ensure dissemination and the best usage of data acquired from public funding, the Neurinfo team strongly encourages investigators to share their data, which is usually agreed after an embargo period.

3.2 The Shanoir@OFSEP Repository

The OFSEP project[**] was selected in response to the national call for projects "Cohorts 2010" as part of the "Investments for the Future" program. This is a collaborative project involving over 40 French expert MS centers. The aim of this project is to build and maintain a nationwide cohort of patients with Multiple Sclerosis (MS), and enrich the clinical data with biological samples, socio-economic data and neuro-images.

A dedicated imaging working group is in charge of acquiring, processing, integrating imaging and derived imaging data into a shared Imaging Resource Centre (IRC), and make this IRC inter-operate with the clinical databases. The consistent assessment of MRI-based measurements at a large scale require robust and efficient image processing pipelines. A further goal of this project is to establish an information technology (IT) infrastructure enabling audited access to imaging data, as well as a "virtual laboratory" environment supporting the distributed, synergistic development, validation, and deployment of specialized image analysis procedures, developed by different national and international research centers. To ensure an easy access to the imaging data and allow modifications, queries, annotations and access control, the Shanoir environment has been selected. It also ensures interoperability and data management related to the imaging part of this cohort (the clinical part is managed by the EDMUS[††] system).

Started in 2012, the Shanoir@OFSEP server has been installed to store the imaging data of the OFSEP cohort. This cohort aims at studying neuroimaging data of 40.000 MS patients over the next 10 years. A consensus has emerged concerning the acquisition protocol that requires: a brain MRI every 3 years, a spinal-MRI every 6 years, that is to say 200.000 MRI over 10 years. Shanoir@OFSEP database will grow during this period and beyond [5].

Since OFSEP is a nationwide project gathering many patients, many IRC and much different MRI equipment, a federated repository with nationwide access and with thorough homogenization mechanism was therefore needed. The OFSEP imaging WG is continuously gathering new acquisition centers volunteering to take part to the cohort. In Shanoir@OFSEP, there are currently about 30 IRCs pooling 40 MRI acquisition equipment representing 12 MR scanner models from 3 MR constructors (Siemens, Philips, GE). All the centers are importing data in one main study called the "Mother Cohort". If necessary, derived imaging data will be then imported back to the server in order to refer to potential post-processing information, MS specific imaging biomarkers making them available for others authorized users.

Currently Shanoir@OFSEP repository is hosting 5 studies: the "Mother Cohort" (200.000 MRI planned over the next 10 years) as well as 4 MS imaging clinical research projects. More of these "OFSEP-labeled" clinical research projects or nested cohorts will

[**] The OFSEP MS Cohort observatory: http://www.ofsep.org/fr/l-observatoire/presentation-ofsep
[††] EDMUS: http://www.edmus.org

be integrated in the following years. Everyone can join the "Mother Cohort" study as long as they use the OFSEP protocol. One can also ask the OFSEP to contribute to the project through his study as soon as the principal investigator presents his research study subject to the OFSEP scientific committee that can grant (or not) the hosting. Data hosted on Shanoir@OFSEP, will remain confidential (private) throughout the duration of the study, but can be made available to all researchers through a specific application to OFSEP.

4 Conclusion and perspectives

The Shanoir Software as a Service environment has been presented. We have shown how this system manages to share distributed information sources in neuroimaging over the Internet, whether these resources are located in various centers of experimentation, clinical departments in neurology, or research centers in cognitive neurosciences or image processing. Through the description of two repositories that administrate a Shanoir environment (Neurinfo and OFSEP), we have illustrated how a large variety of users can diffuse, share or access neuroimaging information between peers almost as easily as if the data were stored on their local hospital, research labs or companies. Through the description of the Shanoir software environment, we have illustrated how neuroimaging data can be structured, managed, archived, visualized and shared with examples on multi-institutional, collaborative research projects.

5 Acknowledgements

This work has been supported by the "technological development program" of Inria, by the Brittany region council and the EU-Feder program for the Neurinfo platform, and by two grants provided by the French Government and handled by the "Agence Nationale de la Recherche," within the framework of the "Investments for the Future" program, under the references ANR-10-COHO-002 (for OFSEP) and ANR-11-INBS-006 (for FLI).

6 REFERENCES

1. Rimal, B.P., E. Choi, and I. Lumb. *A taxonomy and survey of cloud computing systems*. in *INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on*. 2009. Ieee.
2. Michel, F., et al. *Grid-wide neuroimaging data federation in the context of the NeuroLOG project*. in *Proceedings of the HealthGrid*. 2010.
3. Temal, L., et al., *Towards an ontology for sharing medical images and regions of interest in neuroimaging.* J Biomed Inform, 2008. **41**(5): p. 766-78.
4. Barillot, C., et al., *Federating Distributed and Heterogeneous Information Sources in Neuroimaging: The NeuroBase Project.* Stud Health Technol Inform, 2006. **120**: p. 3-13.
5. Cotton, F., et al., *OFSEP, a Nationwide Cohort of People with Multiple Sclerosis: Consensus minimal MRI protocol.* Journal of Neuroradiology, 2015. **(in Press)**.

Population Imaging Study IT Infrastructure: An Automated Continuous Workflow Approach

Marcel Koek[1], Hakim Achterberg[1], Marius de Groot[1], Erwin Vast[1], Stefan Klein[1], and Wiro Niessen[1,2]

[1] Biomedical Imaging Group Rotterdam, Departments of Radiology & Medical Informatics, Erasmus MC, Rotterdam, the Netherlands
[2] Imaging Science & Technology, Department of Applied Sciences, Delft University of Technology, the Netherlands

Abstract. The increasing scale and complexity of population imaging studies poses challenges for managing and maintaining data storage, access and analysis infrastructures. Traditionally the data workflow in a population study is largely managed manually. We propose an automated workflow approach which formalizes and streamlines the management of the image storage and analysis workflow. At the core of the infrastructure is an automated study manager that serves as a mediator between the different components and keeps the master records of the data objects in the study. This approach helps minimize human errors, improve consistency of the data and allows for continuous data flows.

1 Introduction

With advancing scanner technology, wider availability of automated analysis methods and increasingly complex multidisciplinary research questions, data management of population imaging studies is a substantial responsibility. The amount of data a population imaging study generates is increasing and so is the complexity of the data analysis. These large and rich datasets present researchers with great opportunities for new research, but also lead to challenges with respect to the storage and analysis of the imaging data.

Besides the increasing scale of population imaging studies, there are more developments that challenge conventional practices in imaging: (1) Studies acquire more and more non-imaging data and are positioned more in multidisciplinary settings (e.g. imaging genetics), (2) more studies are multicentric, often presenting heterogeneity amongst scanners and scanning protocols, and (3) an increase in imaging biomarkers that are available.

These developments make it more important to manage the data and processing for population studies very carefully. They require robust solutions for storage, processing and management of the study data. For consistency and reproducibility, it is important to automate or otherwise formalize as much of the entire imaging workflow as possible. A complete workflow for every scan should be defined and adhered to, making sure all automatic processing and manual

Marcel Koek, Hakim Achterberg, Marius de Groot, Erwin Vast, Stefan Klein, Wiro Niessen ; Population Imaging Study IT Infrastructure: An Automated Continuous Workflow Approach, In: *Proceedings of the 1st Miccai 2015 Workshop on Management and Processing of images for Population Imaging – MICCAI-MAPPING2015, C . Barillot, M. Dojat, D. Kennedy and W. Niessen (Eds), pp.31-38, 2015.*

actions are performed in the correct way. By minimizing the opportunity for human errors in the workflow, this assures adherence to predefined procedures and provides insight herein for granting bodies, journals and certificate authorities.

To achieve this, it is important to integrate a number of concepts into a comprehensive infrastructure for population imaging. For many imaging infrastructure concepts there are solutions available, but a complete integrated solution is to our knowledge not existing. In this work, we present the development of an infrastructure for population imaging, where we use, as much as possible, existing software. All of the software used is freely available, making it possible for others to replicate (part of) the infrastructure. At the core of this infrastructure a novel component, the study manager, is placed which formalizes and manages the workflow for the study, and automates tasks where possible. It is this study manager that interoperates with, and links together, all systems in the infrastructure. Since parts of the workflow of the population imaging management are automated, the workflow can be executed more continuously.

2 Design

We briefly describe the workflow of existing population imaging studies first. From that we define the requirements for a system that implements solutions for dealing with large amounts of data and high data complexity. Subsequently, we outline the proposed IT infrastructure for population imaging with a central role for the automated study manager.

2.1 Workflow in population imaging studies

Population imaging studies all have a common structure to their workflows. The workflow in most population imaging studies can be split up in roughly the following tasks:

1. Select (new) participants for eligibility
2. Invite eligible participants and schedule scan visits
3. Screen participants for contraindications to scanning
4. Scan participants and gather other information
5. Store all acquired (meta-)data in an archive and/or study database
6. Check scans for incidental findings and major imaging artefacts
7. Retrieve (pseudonymized or anonymized) data for analysis
8. Data analysis at participant level, determining quantitative biomarkers or other statistics
9. Perform a quality assessment of the (processed) data
10. Aggregate level analysis, describing population effects
11. Disseminate results of analysis

All steps in this workflow are manually performed and initiated. The steps are either routinely initiated or per request. While patient inclusion (steps 1-2) and final analysis (steps 10-11) will typically involve groups of participants, the steps in between are typically followed per participant.

2.2 Requirements for a population imaging IT infrastructure

The IT infrastructure is involved in steps 4 through 10 of the workflow described above. The general requirements for a data and analysis management infrastructure, supporting the previously mentioned tasks, for population imaging studies are:

- The data archive must be reliable and recoverable in case of disaster, failures of machines should not corrupt study data integrity
- The state of a scan session in a workflow must be known unambiguously
- As data passes through different tasks in the study workflow, provenance documents must be recorded to document the history of each dataset.
- All components must interact (this requires all core components to have programmatic interfaces)
- The system must be able to accept data continuously and be able to ingest data in batches
- The system needs to remain online, even during routine maintenance
- The system must be able to mix manual procedures with automated processing
- Data on the infrastructure must all be anonymized / pseudonymised
- The system must be scalable. Limiting components need to be parallelizable.

2.3 The proposed infrastructure

The main systems that are used in the proposed population imaging IT infrastructure are:

- An automated study manager for managing the study workflow
- An image archive for storing acquired imaging data and derived imaging data
- A study database for storing subject information and data as well as derived non-imaging data
- An automated image processing engine for managing biomarker generating pipelines
- Viewers and editors for review of incidental findings and quality assessment
- Anonymization / Pseudonymization system

In Figure 1 the infrastructure based on the basic systems is schematically illustrated.

Automated Study manager The study manager keeps track of the state of every scan. It guides the scan sessions through a predefined workflow and is aware of all possible states and transitions in the workflow. This approach offloads administrative overhead and thereby allows for instantaneous processing of data as it comes in, rather than manually starting processing tasks for entire cohorts. A big advantage of this continuous approach is time efficiency. In the automated parts of the workflow there is no need for any buffers where tasks

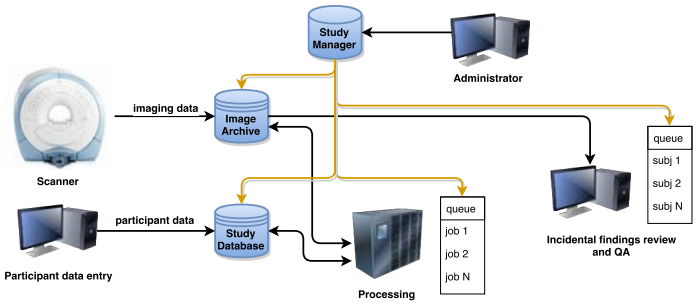

Fig. 1. Schematic overview of the interaction between the basic components of a population imaging workflow. An pseudonymization system could be plugged in between the image archive and the scanner. After processing the results could be reviewed on a workstation.

have to wait until they are executed. This makes it possible to disclose results of the analysis quickly after acquiring the data.

The study manager takes a central place in the infrastructure. It interoperates with every system in the infrastructure. This makes recording high-level data provenance possible. Data provenance records will be created for every scan session for every task in the workflow. Having provenance recorded for every data object in the study is important for reproducibility, for data sharing, or to aid in problem solving. Knowing how a dataset was analyzed is a prerequisite for ensuring the reliability of any analysis.

Every transition in the workflow of the imaging study is recorded. This atomic operation stores everything that is relevant for restoring the situation in case of a system failure. When the study manager needs to recover after a failure it looks for tasks that were pending and requests the current status from the responsible systems in the infrastructure. When the state of a data object is unclear this is reported to an administrator who can take actions accordingly.

Image Archive The image archive holds all imaging data of a population study. To integrate the image archive into the automated workflow infrastructure, it must expose data handling functionality via a programmatic interface. The image archive should support the DICOM[3] image format for raw data, and a selection of other formats for derived data.

Study database All (simple type) data about each participant (e.g. site visit, cognitive data, blood markers, physical tests) are stored in the study database. For an integrated infrastructure, it is important that the database can be queried via a programmatic interface. At the very least, basic demographic information and context for imaging visits should be exposed.

[3] Digital Imaging and Communications in Medicine

Automated image processing engine The image processing engine is responsible for starting and monitoring image processing pipelines. The processing engine should collect data provenance documents for all results generated by the image processing pipeline.

Viewers and editors There are a number of steps that require manual interaction: review for incidental findings, quality assessment of the raw image data and the results of automatic processing, and possibly manual segmentation steps. For this, image viewers and editors are needed that are able to retrieve data from the image archive. Ideally they can also interact with the study manager, so users can easily view/edit the scans that are queued for manual interaction.

Anonymization The proposed infrastructure is designed for research purposes. In general, this means that all data on the infrastructure should be anonymized or pseudonymized. To facilitate easy transfer of data from a clinical partner to the infrastructure, it is important to have a reliable, automated anonymization method. In case the data of a subject is split over multiple independent databases and the results of analysis have to be combined later, the data should be pseudonymized. The pseudonymization keys should be stored with the proper instances at the clinical side.

3 Implementation

This section starts with a description of the study manager. We then describe the existing software solutions available for the other components of the infrastructure. Finally, we briefly describe the current software stack used in our infrastructure.

3.1 Study manager

We are not aware of any existing software that fulfills the specific requirements of the study manager. Therefore, we have started developing a study manager in-house. The study manager basically consists of 3 components, a state machine model, a database and an Application Programmatic Interface (API). Front-ends for managing and monitoring can be made by using the API.

A state machine is a mathematical model of an abstract machine that can have one of a finite number of states. In case of a population imaging study, the study workflow can be modelled by a state machine. Each data object (e.g a scan session) that is part of the study is in a certain state. A data object can be in one state at a time and can only go to a next state when an event or condition triggers a transition. The possible transitions are defined by the workflow. This formalizes the workflow as it enforces the data to flow through the workflow in a predefined order. The workflow is fully customizable so it can fit the needs of

different studies. Figure 2 shows a part of the state machine model for the study workflow.

The state history of every data object in the system is stored in a database. The current state can be extracted by looking for the last state known. The state machine has to be persistent, so that any change is immediately stored in the database and cannot be lost. After a system failure the system needs to be brought in the correct state as soon as possible.

Because other components are not aware of the study manager, the transitions are triggered by meeting certain conditions by default. For every scan session, the study manager checks if the condition of one of the transitions from the current state are met and triggers the transition accordingly. Optionally it is possible to trigger a transition externally via a Representational State Transfer (REST) API [2]. This can be used by administration tools or custom made viewers where the results are reviewed on their quality. The state and basic data of each scan session can be requested via the same REST API.

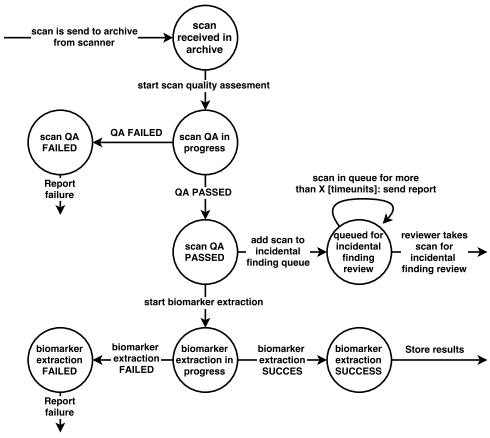

Fig. 2. Part of the state machine model of the study workflow.

3.2 Other components of the IT Infrastructure

For image storage there are a number of possible systems. XNAT[5] is an image archive with a REST interface primarily created neuroimaging studies, but is now used in other imaging domains as well. Midas[4] is an open-source toolkit for creating data storage solutions. SciTran[5] is a project in development, offering a storage solution from imaging data. As an alternative to a dedicated image storage solution, it is also possible to use an object storage such as Openstack

[4] `http://midasplatform.org`
[5] `http://scitran.github.io`

Swift[6], Ceph[7] or a public-cloud based solution and let the study manager interact directly with the storage layer.

There are many study database options, both high level and low level. For clinical research, OpenClinica[8] is a project that offers a study database combined with webforms for convenient data entry. Many population imaging studies use custom made software with database backends to store the study data.

There are many pipeline engines available, most notably: Loni pipeline[1], Nipype[3] and Fastr[9]. They are all created for the domain of medical imaging and are based around the concept of interfacing with command-line tools, without requiring special APIs or recompilation.

For viewing any viewer can be used, for example 3DSlicer[10], ITK-SNAP[11] and FSLView[12] or custom made viewers based on VTK[13], MeVisLab[14], XTK[15] or similar software. For anonymization and routing DICOM files the Clinical Trial Processor has been created by the Radiological Society of North America.

3.3 Currently used components

We currently use XNAT for image archiving, Fastr for managing the analysis, CTP for pseudonymization the data. For reviewing and correction of the analysis results a custom made tool is made using MeVisLab. Incidental finding checking is done on radiological workstations. For some studies OpenClinica is used as the study database, while others have a tailor made study database. Data is stored on a replicated GPFS file system, which offers copy-on-write and atomic snapshots that serve as recovery time points.

4 Discussion

We propose a population imaging infrastructure using a continuous, automated workflow approach. We explored different options for the components required for building this infrastructure. However, the separate components can be interchanged to suit specific requirements of the study and institutional preferences. The study manager acts as mediator between the different components and — most importantly — keeps a record of the state of each individual scan.

As there is no suitable solution for the study manager available, we introduce our own open source software[16]. In our study manager, the transitions between

[6] http://swift.openstack.org
[7] http://ceph.com
[8] https://www.openclinica.com
[9] http://fastr.readthedocs.org/en/default/
[10] http://www.slicer.org
[11] http://www.itksnap.org
[12] http://fsl.fmrib.ox.ac.uk/fsl/fslview
[13] http://www.vtk.org
[14] http://www.mevislab.de
[15] https://github.com/xtk/X
[16] https://bitbucket.org/bigr_erasmusmc/syncotron

tasks are predefined and are automated. However, the tasks themselves can be either manual or fully automated. The automated nature of the study manager avoids human error and ensures consistency of the data. By automating the transitions the workflow becomes continuous, leading to a quick availability of the derived measures. Using predefined transitions ensures the state of a data object is always valid and the state is stored in a persistent database. With this knowledge, the study manager can fully recover after a system crash. While transitioning the data objects (e.g. scan sessions) through the workflow, the study manager records provenance documents.

With this infrastructure proposal we present a blueprint for a continuous, automated management of population image studies. Our design is based on our experience with the Rotterdam Scan Study which contains more than 12000 scan sessions of over 5800 unique participants at time of writing[4]. It is designed to handle the increasing scale of population imaging studies, the complexity of analysis methods, and the growing number of automatically derived biomarkers.

5 Acknowledgements

This work was supported by the following projects: BioMedBridges[17], CTMM TraIT[18], CVON Heart Brain Connection[19], European Population Imaging Infrastructure[20] and BBMRI-NL2.0[21].

References

1. Dinov, I., Lozev, K., Petrosyan, P., Liu, Z., Eggert, P., Pierce, J., Zamanyan, A., Chakrapani, S., Van Horn, J., Parker, D.S., et al.: Neuroimaging study designs, computational analyses and data provenance using the loni pipeline. PloS one 5(9), e13070 (2010)
2. Fielding, R.T., Taylor, R.N.: Principled design of the modern web architecture. In: Proceedings of the 22Nd International Conference on Software Engineering. pp. 407–416. ICSE '00, ACM, New York, NY, USA (2000)
3. Gorgolewski, K., Burns, C.D., Madison, C., Clark, D., Halchenko, Y.O., Waskom, M.L., Ghosh, S.S.: Nipype: a flexible, lightweight and extensible neuroimaging data processing framework in python. Frontiers in neuroinformatics 5 (2011)
4. Ikram, M.A., van der Lugt, A., Niessen, W.J., Krestin, G.P., Koudstaal, P.J., Hofman, A., Breteler, M.M., Vernooij, M.W.: The rotterdam scan study: design and update up to 2012. European journal of epidemiology 26(10), 811–824 (2011)
5. Marcus, D.S., Olsen, T.R., Ramaratnam, M., Buckner, R.L.: The extensible neuroimaging archive toolkit. Neuroinformatics 5(1), 11–33 (2007)

[17] `http://www.biomedbridges.eu`
[18] `http://www.ctmm-trait.nl`
[19] `http://www.heart-brain.nl`
[20] `http://populationimaging.eu`
[21] `http://www.bbmri.nl`

Fastr: a workflow engine for advanced data flows

HC Achterberg[1], M Koek[1], and WJ Niessen[1,2]

[1] Biomedical Imaging Group Rotterdam, Depts. of Radiology & Medical Informatics
Erasmus MC, Rotterdam, the Netherlands
[2] Imaging Science & Technology, Faculty of Applied Sciences
Delft Univ. of Technology, the Netherlands

Abstract. With the increasing number of datasets encountered in imaging studies, the increasing complexity of processing workflows, and a growing awareness for data stewardship, there is a need for managed, automated workflows. In this paper we introduce Fastr, an automated workflow engine with support for advanced data flows. Fastr has built-in data provenance for recording processing trails and ensuring reproducible results. The extensible plugin-based design allows the system to interface with virtually any image archive and processing infrastructure. This workflow engine is designed to consolidate quantitative imaging biomarker pipelines so that they can easily be applied to new data.

1 Introduction

In medical image analysis, most methods are no longer implemented as a single computer program, but as a comprehensive workflow composed of multiple programs that are run in a specific order. Each program is executed with inputs that are predetermined or are following from the results of previous steps. With increasing complexity of the methods, the workflows become more convoluted and encompass more steps. This makes execution of such a method by hand tedious and error-prone and has led to solutions based on scripts that perform all the steps in the correct order.

In population imaging, data collections are typically very large and are often acquired over prolonged periods of time. As data collection is going on continuously, the concept of a 'final' dataset is either a non-existent or at least a far away time point. Commonly, analyses on population imaging datasets therefore define intermediate cohorts or time points. All image analysis methods need to produce consistent results over time and should be able to cope with the ever growing size of the population study. Therefore the process of running analysis pipelines on population imaging data needs to be automated to ensure consistency and minimize errors.

Traditionally this is accomplished by writing scripts created specifically for one processing workflow. This can work well, but generally the solutions are tailor-made for a specific study and software environment. This makes it difficult to apply such a method to different data or on different infrastructure than originally intended. With evolving compute resources, in practice this approach

Hakim Achterberg, Marcel Koek, Wiro Niessen ; Fastr: a workflow engine for advanced data flows, In: *Proceedings of the 1st Miccai 2015 Workshop on Management and Processing of images for Population Imaging* – MICCAI-MAPPING2015, C . Barillot, M. Dojat, D. Kennedy and W. Niessen (Eds), pp.39-46, 2015

is therefore not reproducible and difficult to maintain. Additionally, for transparency and reproducibility of the results it is very important to know exactly how the data was processed. To accomplish this, an extensive data provenance system is required.

Writing a script that takes care of all the aforementioned issues is a challenging and time consuming task. However, many of the components are generic for any type of workflow and do not have to be created separately for each workflow. Therefore, we developed an image processing workflow framework for creating and managing processing pipelines: Fastr. The framework is designed to build workflows that are agnostic for where the input data is stored, where the resulting output data should be stored, where the steps in the workflow will be executed, and what information about the data and processing needs to be logged for data provenance. To allow for flexible data handling, the input and output of data is managed by a plugin-based system, which allows for flexible data handling. Where and in what order the workflow steps are executed is managed by a pluggable system as well. The provenance system is a built-in feature that ensures a complete log of the processing.

2 Design

In the Fastr philosophy, a workflow is built from a number of atomic steps. Each step can be considered a call to a command line executable. Each different executable that can be called by the system we call a tool. Each tool needs data and/or parameters (input) to process and generate output data. These inputs and outputs are fed to the executable via command line arguments. Fastr needs to be made aware of how to feed data to the tool and how to get data from the tool. Therefore, the inputs and outputs of each tool along with information about where and how to locate the executable are described in simple xml files.

Once the required tools are known to the system, a workflow can be created. A workflow in Fastr is represented by a network of nodes. A node is a step in a workflow and can be considered as an instantiation of a tool. Figure 1 shows a simple graphic representation of an atlas-based segmentation workflow, using the Elastix registration package[4]. In Elastix, a deformation field is optimized to match the moving image to the fixed (reference) image. There are different classes of nodes: normal nodes, source nodes, constant nodes and sink nodes. The source nodes and constant nodes are places where data can enter the network, whereas sink nodes are the places where the data can leave the network. Data flow in the network is defined by links. A link is a connection between the output of a node and the input of another node. The nodes and links in the network form a graph from which the dependencies can be determined for the execution order.

2.1 Data model

Data in Fastr is represented by samples. A sample is the unit of data that is presented to inputs of a node for a single run. It can be a simple scalar value,

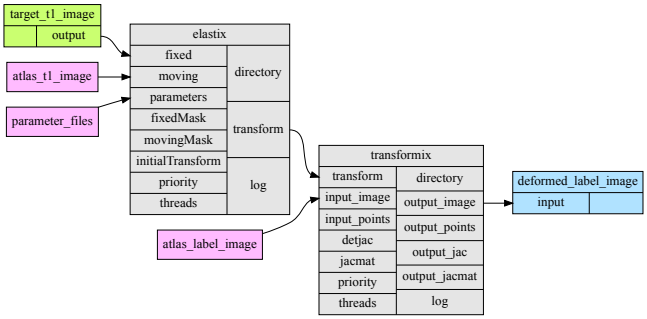

Fig. 1. Example Network representing a single atlas-based segmentation workflow implemented using the open source Elastix image registration software. Green boxes are source nodes, purple constant nodes, gray normal nodes, and blue sink nodes. Each node contains two columns: the left column represents the inputs,the right column represents the outputs of the node. The arrows indicate links between the inputs and outputs. This image was generated automatically from the source code.

a string, a file, or a list of these. To illustrate this, in this section we will use the Elastix Node from 1 as an example Node. The fixed and moving inputs of the Elastix node are required to be images. The parameters input should be supplied with one or more text files defining the registration settings; therefore a sample of this input should be a list of parameter files. The transform output will generate a sample that contains a list of transform files (one transform file for each input parameter file). The amount of values a sample contains is called the cardinality of the sample.

Figure 2 illustrates a number of situations where samples are offered to inputs (f for fixed, m for moving, and p for parameters) that results in a number of samples (t for transforms). In Figure 2a we present the simplest situation, where one sample with one value is offered to each input and one sample with one value is generated. In Figure 2b, the fixed and moving inputs have one sample with one value, but the parameters input has one sample with two values. The result is a sample with two values, as one transform file is created per parameter file.

To facilitate batch processing a node can be presented with a collection of samples. These collections are multi-dimensional arrays of samples. In Figure 2c, we depict a situation where three registrations are performed. Three samples are offered to the fixed input and one sample is offered to the other inputs. This results in three samples: each sample of the fixed input was used in turn, whereas the samples for the moving and parameters were considered constant. In Figure 2d, there are three samples for the fixed and moving inputs. The result is again three samples as now each pair of samples from fixed and moving inputs was taken, and the parameters was considered constant.

This is useful for simple batch processing where a task should be repeated a number of times for different input values. However, in a multi-atlas based segmentation for example, it is required to register every fixed image (the targets)

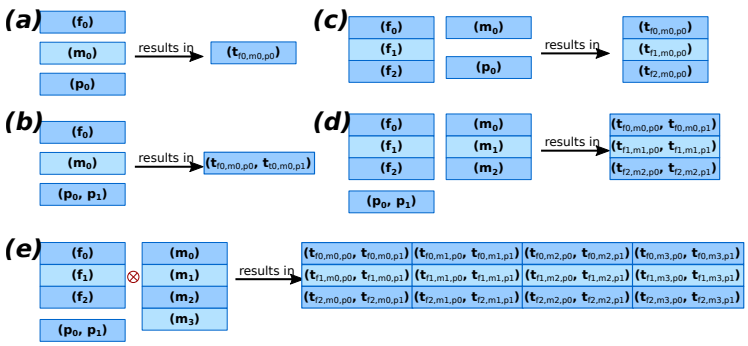

Fig. 2. Illustration of the data flows in a Node. Each rectangle is a sample, and a block of rectangles represents a sample collection. The value is printed in each rectangle, where the commas separate multiple values. The samples f are offered to the fixed input, the sample m to moving input and the sample p to the parameters input. The sample t are generated for the transform output. The sample t subscript indicates which input samples were used to generate the result.

to every moving image (the atlases). To simplify this procedure Fastr can switch from pairwise behaviour to an outer product behavior. In Figure 2e, this is depicted graphically. Every combination of fixed and moving sample is used for registration and the result is a two-dimensional array of transformation samples that in turn contain two transformations each.

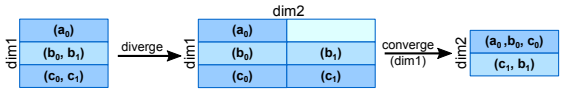

Fig. 3. Converging and diverging flows. The start situation on the left diverges to the situation in the middle after which data converges the first dimension. Note that in the middle situation there is an empty place in the sample collection (top right). This is possible due to a sparse array representation of the sample collections. This results in two samples with different cardinality in the right-most situation.

Sometimes a tool outputs a sample with a higher cardinality which should be treated as separate samples for further processing, or conversely a number of samples should be offered as a single sample to an input (e.g. for taking an average). For this, Fastr offers two flow directives in data links. The first directive is diverging which indicates that the cardinality is to be transformed into a new dimension. This is illustrated in the left side of Figure 3. The second directive is converging which indicates one or more dimensions in the sample array should be collapsed and combined into the cardinality. This process is illustrated in the right side of Figure 3. These flow directives allow for more complex dataflows in a simple fashion and enable users to implement MapReduce type of workflows.

2.2 Data input and output

The starting points of every workflow are source nodes, in which the data is imported into the networks. Similarly the endpoints of every workflow are the sink nodes which export the data to the desired location. When constructing a network, the sources and sink need to be defined, but the system only needs to know the type of data that will be presented. The actual definition of the data is done at runtime using uniform resource identifiers (URI).

Based on the URI scheme, the retrieval/storage of the data will be performed by a plugin. Given two example URI:

```
vfs://mount/some/path/file1.txt
xnat://xnat.example.com/data/archive/projects/sandbox/subj...
```

The schemes (in red) of these URIs are different. This means that the first URI will be handled by the Virtual File System plugin, whereas the second URI will be handled by the XNAT[3] storage plugin. These plugins implement the methods to actually retrieve and store the data. The remainder of the URI is handled by the plugin, so the format of the schemes URI format is defined by the plugin developer.

There are also plugins that can expand a single URI into multiple URIs based on wildcards or searches. In the following example URIs we use wildcards (shown in blue) to retrieve multiple datasets in one go:

```
xnat://xnat.example.com/search?project=test&subjects=s[0-9]...
vfsregexp://tmp/network_dir/.*/.*/__fastr_result__.pickle.gz
```

The XNAT storage plugin has a direct storage as well as search URI scheme defined. The VFS regular expression plugin, uses the regexp filter to generate a list of matching vfs URIs.

This makes the network completely agnostic to the location and storage method of the source and target data. Also it allows easy loading of large amounts of resources using wildcards, csv files or searches. Currently Fastr includes plugins for input/output from the (virtual) filesystem, csv files and XNAT.

2.3 Execution

The Fastr framework is designed to offer flexible execution of jobs. The framework analyzes the workflow and creates a list of jobs, including dependencies, that need to be executed. Then it dispatches the jobs to an execution plugin. A different plugin can be selected for each run allowing for easy switching of the execution backend. The plugins can dispatch jobs to an execution system such as a cluster, grid, or cloud.

The Fastr execution system carries out the following steps:

- The Network is analyzed and an ordered list of nodes is generated; at this stage, logical errors in the network will be identified and thrown
- A job for each desired sample combination is created for each node
- The jobs are dispatched by the execution plugin
- The execution plugin schedules and runs the job

– The job executes the actual processing, gathers and validates the results and creates a provenance record.
– The execution plugin sends a callback to the network to process the new data

Currently, Fastr supports functional plugins for processing locally and on a cluster (using the DRMAAv1 api[3]). Next plugins will focus on flexible middleware for grid/cluster/cloud, like Dirac[4], that offer support for a wide range of systems.

2.4 Provenance

Data provenance is a built-in feature of Fastr. An implementation of the W3C Prov Data model (PROV-DM) is used to provide this. Fastr records all relevant data during execution and ensures that for every resulting file from a sink a complete data provenance document is included. The standard format of a provenance document is PROV-N, which can be serialized to PROV-JSON or PROV-XML.

In Figure 4 the three base classes and the properties of how they relate to each other are illustrated. For Fastr, networks, tools and nodes are modelled as agents, jobs as activities and data objects as entities. The relating properties are naturally valid for our workflow application. The hierarchy and topology of the network follows automatically from the relating properties between the classes, but in order to make the provenance document usable for reproducibility, extra information is stored as attributes on the classes and properties. For every agent in our system the version is stored and for every entity the value or file path and an md5sum is stored. For every activity the start and end time of execution, the stdout and stderr logs are stored, the end status (success, success with warnings, failed, etc), and an exhaustive description of the environment.

3 Results

A functional version of Fastr is available from `https://bitbucket.org/bigr_erasmusmc/fastr`. It is released under the Apache license 2.0, which means it is open-source and free to use. The framework is written in Python and easy to install using the standard Python tools. Fastr is platform independent and runs on Linux, Mac and Windows environments. However the focus lies on supporting Linux since that is the platform used in most processing environments.

The project has online documentation at `http://fastr.readthedocs.org`. It includes a small tutorial, a user manual and a developer reference of the code built using Sphinx.

Currently we are using the system for a number of workflows for multiple large studies as well as some projects with other departments in-house. For example,

[3] `http://www.drmaa.org`
[4] `http://diracgrid.org`

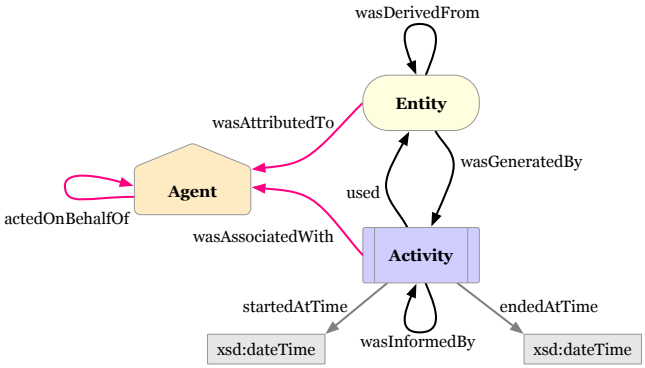

Fig. 4. The three base classes of the provenance data model with their relating properties. The agents are orange pentagons, the entities are yellow ovals and the activities are depicted as blue squares. This image is copied from PROV-O: The PROV Ontology. Copyright 2015 W3C (MIT, ERCIM, Keio, Beihang).

the Rotterdam Scan Study uses a Fastr workflow for the preprocessing, tissue type segmentation and lobes segmentation of brain images. The data is fetched from the archive and is processed in a cluster environment. The resulting data is stored in an image archive.

4 Discussion

With Fastr we created a workflow system that allows users to rapidly create workflows. The simple access to advanced features makes Fastr suitable for both simple and complex workflows. Workflows created with Fastr will automatically have a good provenance system, support for execution on various computational resources, and support for multiple storage systems. Therefore, Fastr speeds up the development cycle for creating workflows and minimizes the introduction of errors.

There are other workflow systems developed for the domain of medical image analysis. Most notable are LONI pipeline and Nipype. LONI pipeline[1] is a mature package with a GUI to create workflows for neuroimaging. This package is maintained by the LONI group at the University of Southern California and provided as a closed-source package. Though this system works well in the LONI infrastructure, the close-source nature makes it more difficult to fit into different infrastructures and extend the system with new features.

The only open-source, domain-specific tool that we are aware of is Nipype[2], which is aimed at creating a common interface for varying neuroimaging tools. It also features a system for creating workflows. Compared to Fastr, the tool interfaces Nipype uses are slightly more labour intensive to create, but more advanced and flexible due to the extra freedom. Their design allows for easy manual exploration of data and prototyping of processing. Conversely, Fastr is

more aimed at large scale batch processing and managed pipelines. The Fastr data model is inherently aimed at batch processing and advanced data flows, whereas Nipype uses MapNodes and JoinNodes for non-linear data flows. Also, Fastr allows for multiple versions of tools to be available simultaneously, whereas Nipype by default searches for executables on the PATH. We believe it is important to be able to keep an environment where all the old versions of tools are available for future reproducibility of the results.

4.1 Future directions

As Nipype and Fastr both offer distinct powerful features, it might be worthwhile to interchange ideas with Nipype. Considering that there are many interfaces available for Nipype, we want to support Nipype interfaces with the Fastr workflow system in the future. However, it is not trivial to combine the Nipype interfaces with the Fastr tool versioning principle. It would also be very advantageous to share the provenance definition used, so that both platforms can append to each others provenance in a natural fashion.

For reproducibility it is important to be able to re-run analyses in exactly the same conditions. Currently Fastr supports environment modules to keep multiple versions of software available at the same time. However, the same version of the software can still be different based on underlying libraries, compiler used and the OS. Docker[5] offers a solution to this problem. Docker is a lightweight linux container, that works similar to virtual machines, but is much more efficient. It ensures that the binaries and underlying libraries are all the same between runs. We plan to add support for Docker containers to make it easier to share tools and improve reproducibility further.

Finally, we are working on more (web based) tooling around Fastr to make it is easier to visualize/develop networks and inspect the results of a run (including provenance information).

References

1. Dinov, I., Lozev, K., Petrosyan, P., Liu, Z., Eggert, P., Pierce, J., Zamanyan, A., Chakrapani, S., Van Horn, J., Parker, D.S., et al.: Neuroimaging study designs, computational analyses and data provenance using the loni pipeline. PloS one 5(9), e13070 (2010)
2. Gorgolewski, K., Burns, C.D., Madison, C., Clark, D., Halchenko, Y.O., Waskom, M.L., Ghosh, S.S.: Nipype: a flexible, lightweight and extensible neuroimaging data processing framework in python. Frontiers in neuroinformatics 5 (2011)
3. Marcus, D.S., Olsen, T.R., Ramaratnam, M., Buckner, R.L.: The extensible neuroimaging archive toolkit. Neuroinformatics 5(1), 11–33 (2007)
4. Stefan Klein and Marius Staring and Keelin Murphy and Max A. Viergever and Josien P.W. Pluim: elastix: a toolbox for intensity-based medical image registration. IEEE Transactions on Medical Imaging 29(1), 196 – 205 (January 2010)

[5] `https://www.docker.com`

Design and implementation of a generic DICOM archive for clinical and pre-clinical research

Julien Lamy[1], Romain Lahaxe[1], Jean-Paul Armspach[1], and Fabrice Heitz[1]

ICube, Université de Strasbourg, CNRS (UMR 7357),
300 bd Sébastien Brant, 67412 Illkirch Cedex, France

Abstract. DICOM is today an ubiquitous standard for medical imaging, used by an overwhelming majority of modalities and archives deployed in radiology departments. Earlier versions of DICOM were centered on clinical practice, though later versions introduced concepts related to animal imaging and clinical trials: the current standard version has all the specifications required to create an archive of DICOM images for clinical and pre-clinical research.
However, several prominent archives geared towards clinical research have their roots in clinical practice, implying a small set of highly-controlled image sources, which makes them ill-suited when used in the much more chaotic environment of research. In this paper, we present the design and implementation details of a generic DICOM-based picture archive for clinical as well as pre-clinical research. This design includes data normalization, modality-independent storage, generic queries and fine-grained access control rules.

1 Introduction

Since its inception in the early 1990s, DICOM has become a world-wide standard for a wealth of activities related to medical imaging: from large image archives for a whole hospital to display and worklist management, it covers imaging not only of human subjects, but also of animals and tissue samples. Notwithstanding the inclusion of data model for clinical trials only in recent versions of the standard, this ubiquity in clinical practice is however not mirrored in clinical research.

Although seemingly transparent, the switch from clinical practice to clinical research is not an easy one: the data used in clinical trials is increasingly multi-centric in order to reach a better statistical power and with this variety of sources comes a wide variability regarding not only the image content – such as signal-to-noise ratio or voxel size – but also the metadata attached to the image. Prominent softwares such as DCM4CHEE [8], Shanoir [7] and XNAT [9] are widely and successfully deployed when medical imaging archives for clinical research are needed, but in our opinion, they still lack the flexibility required for smooth operation when storing multi-modal as well as human and animal images.

When building a picture archive for clinical research, three main points are to be considered: the preprocessing of data before storage (since the data comes

Julien Lamy, Romain Lahaxe, Jean-Paul Armspach, Fabrice Heitz ; Design and implementation of a generic DICOM archive for clinical and pre-clinical research, In: *Proceedings of the 1st Miccai 2015 Workshop on Management and Processing of images for Population Imaging* – MICCAI-MAPPING2015, C . Barillot, M. Dojat, D. Kenn y and W. Niessen (Eds), pp.47-54, 2015.

from heterogeneous, external sources), the query capabilities (what stored elements can be queried or filtered), and the access control (which operation each person is allowed to perform).

Imaging data in clinical research has two main specificities when compared to clinical practice: heterogeneity and a requirement for anonymity. The heterogeneity can appear either through the names of objects (subjects, time points, MR sequence names, etc.) or through the DICOM elements used to encode a given information: even though the elements concerning diffusion MRI (e.g. gradient direction and b-value) have been specified by the DICOM standard since 2003, we have yet to receive images where those informations are not stored in vendor-specific fields. Concerning the naming of objects, different acquisition centers will have different naming conventions, and, even though this seemingly only affects post-hoc studies, it is quite common in our experience for acquisition centers to disregard the naming guidelines of imaging protocols; in a particularly severe example we encountered, the data from a trial with 160 subjects from 29 different centers contained 61 different series names for a classical T1 3D MR sequence. A similar problem arises when considering anonymity: due to multiple factors, human as well as technical, the subject naming rules given by the principal investigator might not hold, and data might be either nominative or contain wrong identifiers. In most cases, new elements will also need to be added to the data, to accommodate the clinical research model (acquisition center, time points, etc.) as well as archive-specific elements.

Data storage must allow more generic queries than the simple patient / study / series discovery: it is common to perform queries based on acquisition date (e.g. for quality control), subject demographics (e.g. to study sub-groups based on age), or acquisition parameters (e.g. to study the effect of MR sequence parameters on image quality). Due to the large number of existing elements in DICOM data and the ever-evolving nature of medical imaging which translates to an ever-evolving DICOM standard, it is not possible to predict every query that the users of an archive could perform: any database storing medical images should allow generic queries and be future-proof, i.e. it should not require a change of schema anytime the DICOM standard evolves. Since research data is less static than clinical data – errors and ensuing normalization being possibly discovered a long time after the initial storage – the storage should also allow traceability, to guarantee that any modification to the original data has a known author, date and reason, and can be reversed.

In the context of a large archive, users will have different access rights according to the study they request and to the operation they want to perform: in a typical environment, principal investigators will have extended permissions (read and write) on the studies they lead but should have no rights to other studies, while clinical research associates will be authorized to submit specific data (e.g. only a specific MR sequence) and image processing specialists will only have read permissions. This variety of roles shows the need for a fine-grained access control, depending not only on the user identity, but also on the type of

operation performed (i.e. the DICOM service invoked) and the content of the operation data (i.e. the content of the query or of the submitted data).

In the following sections, we will detail how we designed a system meeting all these requirements, along with implementation details.

2 Design

2.1 Conversion, normalization and de-identification

The pre-processing of data before storage, consisting mainly of conversion, normalization, and de-identification operations, can be modeled generically as a routing system – similar to the usual concept used by computer networks – based on DICOM files.

A routing rule can be seen as a pair of a condition and a list of actions: each piece of data entering the system is checked against the condition, and should the condition be met, actions are applied sequentially. Actions can be atomic (e.g. check the existence or value of an element) or composite (logical negation, conjunction or disjunction).

Working on field-based data gives us three main operations – field modification, addition or deletion – where the new value in the case of modification or deletion can be either static or dynamically depending on the values of other fields. Data normalization will typically involve mostly field modification using value mappings (e.g. for the sequence name), while the de-identification profiles specified by the DICOM standard will use mostly deletions (e.g. postal address of the patient) and modifications based on dynamic values (e.g. replace the patient's birth date with the patient's age). Figure 1 shows an example of such a rule.

Under these definitions, a routing system is composed of a list of rules, applied sequentially. The conditions and actions of each rule in a routing system can be arbitrary, but, for easier comprehension, should remain disjoint.

2.2 Storage

The different data models proposed by the DICOM standard are all entity-relationship models. The storage components of several prominent medical image archives (DCM4CHEE, Shanoir, XNAT) follow this model and store the data that can be queried in an SQL-powered relational database. Although this seem

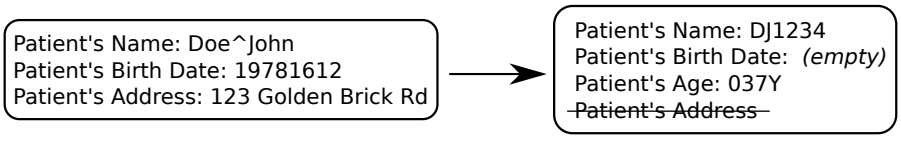

Fig. 1. Example of DICOM routing

the logical choice for an entity-relationship model, the SQL-based implementations are very limited in term of search functionality: due to the large number of elements (over 3800 in the current version of DICOM) and the wealth of IODs and modules, a complete SQL-based backend would yield an intractable number of tables.

In our opinion, the constant nature of an SQL schema is ill-suited to the storage of medical data, especially when storing multi-modal data from both human and animal subjects, since there will be a wide variety of elements from one data set to another. In the last decade, new database mechanisms have appeared, gathered under the "NoSQL" [5] umbrella term: used by Big data actors – such as Google, Amazon or Facebook – NoSQL databases include document-oriented databases, where the data structuration is less strict than in SQL. Under this model, each DICOM data set is represented as a document structured by its elements. The fields of the data set stored in the database are user-definable, ranging from a simple system containing only basic information (patient's name, study and series dates and description), to a complete indexation of the data set, including an interpretation of known vendor-specific fields (e.g. those containing DTI information). The field-based nature of DICOM documents is especially well-suited to document-oriented databases, and even more so since the latest iterations of the DICOM standard specifies XML and JSON models for data sets, two technologies abundantly used in document-oriented databases.

Along with the metadata, some document-oriented database engines allow storing the original data set itself in the database instead of storing it directly on the file system: DICOM files are typically small (from a few hundred kilobytes for 2D files to a few tens of megabytes for usual 3D files), and their number can hence quickly grow to several millions for a medium-sized archive; devising a naming scheme which prevent collisions for that many files is no easy task, and leveraging the database engine to store the original file relieves the developers of such a burden. Moreover, since document-oriented databases originate in the world of Big data, most of them have built-in scaling capabilities [4], allowing the administrator to balance the load on medium-sized or even convenience servers, growing the cluster along with the archive. Storage details, specifically according to the file size, will be given in Section 3.

2.3 Access control

Since DICOM specifies how the user authentication must be performed, using either a user/password, a Kerberos ticket or a SAML assertion, we will assume that the user identity has been verified, and focus on the access control.

As mentioned in the introduction, users will have different roles, not only according to their identity, but also according to the study to which they want to submit data or from which they want to retrieve data. More specifically, an operation is granted or denied access based on the user identity, the service they are accessing, and the data related to the service, either in the query or in the response.

Based on the standardized DICOM service classes, we identified four different services on which to specify access control: echo (reachability of the archive), store, query and retrieve. Since medical data must be protected, even when anonymized, the query service is separated from the retrieve service: a user may have the permission to know the existence of some data, but be required to ask a specific authorization (e.g. approval by an ethics committee) before actually accessing it.

Apart from the echo service, which has no associated data, the request and response data will allow the fine tuning of the permissions, allowing a user to only read from a specific trial, or to only submit a specific modality (e.g. MR, but no PET). Wide-ranging permissions, either on the service type or on the associated data, can of course be specified for trusted users.

This access control scheme is applicable on the original DICOM services (C-FIND, C-STORE, C-GET, etc.) as well as on their equivalent web services (QIDO, STOW, WADO), since the user identity is conveyed in the HTTP headers.

3 Implementation

Our implementation mostly uses C++, leveraging the constructs included in C++ 11 [2], such as type inference and lambda functions, and is available on GitHub [3]. We have split the system in two main projects, called *dicomifier*[1] for the routing part, and *dopamine*[2] for the archive.

3.1 Routing

Since data sources are heterogeneous, a routing system must provide easy to write rules so that the time spent by operators or administrators remain minimal. We have chosen to use an XML representation for the rules, since it allows a standardized serialization, a direct access for advanced users, and an easily-implementable GUI for less advanced users. The following shows an example of an XML rule which converts the patient's birth date to his age and empties the birth date element.

```
<Rule>
    <Condition>
        <ElementMatch tag="PatientBirthDate" value="?*"/>
    </Condition>
    <Action>
        <AgeFromBirthDate />
        <EmptyElement tag="PatientBirthDate"/>
    </Action>
</Rule>
```

[1] https://github.com/lamyj/dicomifier
[2] https://github.com/lamyj/dopamine

We have used the same principle of rules to convert Bruker data sets, our main source of data for small animal MRI to DICOM: each documented Bruker field is mapped, either directly or after transformation, to a DICOM field. The resulting DICOM data set can then be injected in the rest of the routing pipeline, for potential normalization and storage.

3.2 Storage and access control

We have chosen to use MongoDB [6] as our database backend. MongoDB is a stable software with a well-documented C++ API and a native document representation using the BSON [1] format. It is designed to be scalable, by distributing the data across several machines, and redundant, by transparently replicating the data.

As its name suggests, the BSON format is a binary version of JSON, and, since the JSON representation of data sets is specified by the DICOM standard, the transformation of a data set to a document is straightforward, with two specificities. First, we exploit the binary nature of BSON by storing binary fields – i.e. those with a value representation of OB, OF, OD, OW or UN – directly in the document instead of encoding them in Base64. This yields a smaller footprint, since Base64 encoding incurs a size overhead of 33 %. Second, all elements that may contain non-ASCII characters – i.e. all strings except those with a VR of AE or CS – have their value re-encoded as UTF-8. This way, the search operations are independent from the value of the Specific Character Set element.

We then offer the possibility to filter the elements from the data set that are stored in the document: it is not desirable to store all elements in the documents, since some elements are not usable for search purposes. This is the case for vendor-specific elements with unknown semantics and for long elements such as Pixel Data. Indices for the data set collections are configurable and default to the unique identifiers and the descriptions at the patient, study, and series levels.

MongoDB proposes two different solutions for file storage: either in the document itself for small files, or using a virtual file system called GridFS. GridFS simply splits the file into small chunks – each one weighting 255 kB at the time of writing – and store chunk information and file metadata in two collections. Following MongoDB's guidelines, we only store large data sets in GridFS, while smaller data sets are store directly in the document. The cut-off point is 16 MB, the maximal document size allowed by MongoDB at the time of writing: this minimizes the complexity of the request when sending a file, since most of the data we store is still 2D DICOM files, weighting a few hundreds kilobytes.

Concerning the access control, the authentication part is handled by a simple plugin system, allowing administrators to choose the source of identity validation. We supply simple examples for password file and LDAP authentication, enabling local as well as centralized authentication (e.g. using Active Directory). The access control rules are simply stored in MongoDB, using a simple document schema. Any operation not matching an access control item is denied access.

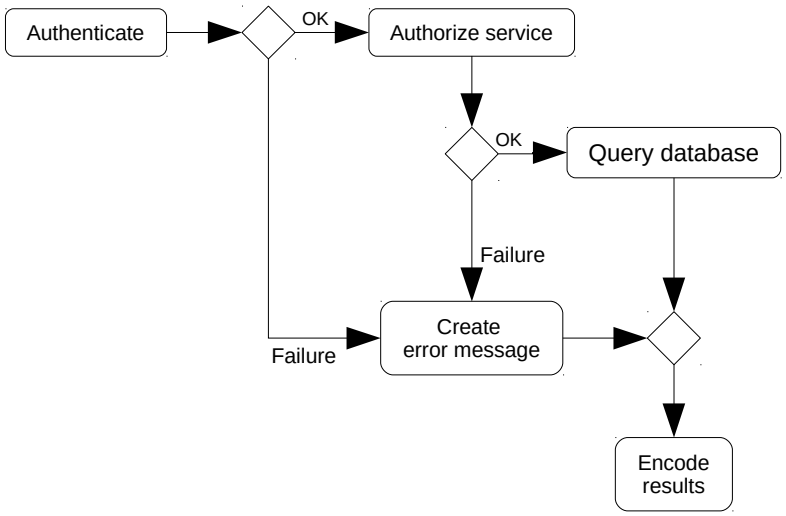

Fig. 2. Data flow

In the following example, the user *scott* can only retrieve images from the patient *Doe^John*, while the user *dr_bob* can perform any operation :

```
{ user: "scott", service: "Retrieve", dataset: ["00100010": "Doe^John"]},
{ user: "dr_bob", service: "*"}
```

Figure 2 summarizes the data flow in the system, for both original DICOM services (C-STORE, C-FIND, C-GET, etc.) and newer web services (STOW, QIDO, WADO).

4 Conclusion

We have presented in this paper a DICOM picture archive able to store multi-modal DICOM data originating from either human or animal subjects. This software allows normalization of data sets before storage, generic queries, and fine-grained access control rules to tailor the privileges according to the user identity. We have deployed this software as our primary archive for animal images, and plan to extend its use to our human images, acquired either in-house or from external sources.

Looking back at the first months of use, we intend to develop the following three new features. First, while the XML routing rules are not difficult to write for a developer, this poses an acceptance problem for non-technical users: we need to develop a simple GUI to generate and edit those rules. Second, the data in clinical research is less static than the data in clinical practice: post-hoc

modification is common, as is storing undesired data (e.g. failed segmentations); to solve this, we plan to specify and implement new DICOM services to modify and delete data, keeping of course the traceability of modifications and enforcing the access control rules for both modification and deletion. Last, we would like to integrate our archive to well-known image processing software (e.g. FSL, SPM, FreeSurfer), so that users could run their preferred algorithms offline on whole cohorts, leveraging the high-performance computing resources of their institutions, thus creating reproducible studies, where each pipeline step could be validated, traced, and replayed when necessary.

References

1. BSON. http://bsonspec.org/
2. Programming language: C++. International Organization for Standardization (2011)
3. GitHub. https://github.com/
4. Hecht, R., Jablonski, S.: NoSQL evaluation: A use case oriented survey. In: Cloud and Service Computing (CSC), 2011 International Conference on. pp. 336–341 (2011)
5. Jing, H., Haihong, E., Guan, L., Jian, D.: Survey on NoSQL database. In: Pervasive Computing and Applications (ICPCA), 2011 6th International Conference on. pp. 363–366 (2011)
6. MongoDB. https://www.mongodb.org/
7. Shanoir. http://www.shanoir.org/
8. Warnock, M.J., Toland, C., Evans, D., Wallace, B., Nagy, P.: Benefits of using the DCM4CHEE DICOM archive. Journal of Digital Imaging 20, 125–129 (2007)
9. XNAT. http://www.xnat.org/

MAPPING-2015, 1st MICCAI Workshop on Management and Processing of images for Population Imaging

MICCAI-MAPPING2015

https://project.inria.fr/fli/mapping-workshop /