Shanoir NG – Dcm2Nii

Software Design Description

Plan

[1 Microservice context 3](#_Toc474167147)

[1.1 Microservice presentation 3](#_Toc474167148)

[1.2 Main functionalities 3](#_Toc474167149)

[1.3 Application users 3](#_Toc474167150)

[2 Database architecture 3](#_Toc474167151)

[2.1 Nifti Converter 3](#_Toc474167152)

[2.1.1 Name 3](#_Toc474167153)

[2.1.2 Is Active 3](#_Toc474167154)

[2.1.3 Ref Nifti Converter Type Id 3](#_Toc474167155)

[2.1.4 Comment 3](#_Toc474167156)

[2.2 Ref Nifti Converter Type 3](#_Toc474167157)

[2.2.1 Label Name 3](#_Toc474167158)

[2.3 Study Card 3](#_Toc474167159)

[2.4 Dataset Expression 3](#_Toc474167160)

[3 Front-end architecture 4](#_Toc474167161)

[3.1 Manage Nifti Converters 4](#_Toc474167162)

[4 Back-end architecture 4](#_Toc474167163)

[4.1 4 Scenarios for nifti conversion: 4](#_Toc474167164)

[4.1.1 Import 4](#_Toc474167165)

[4.1.2 View Dataset Detail 4](#_Toc474167166)

[4.1.3 Manage Datasets 4](#_Toc474167167)

[4.1.4 View Detail Study tree 4](#_Toc474167168)

[4.2 The conversion steps in details 4](#_Toc474167169)

[4.2.1 For the dicom import process: 6](#_Toc474167170)

[4.2.2 To (re)generate nifti files for dataset(s): 6](#_Toc474167171)

[4.2.3 The convertToNiftiExec function 7](#_Toc474167172)

[4.2.4 The post-conversion 7](#_Toc474167173)

[4.3 The conversion black box 8](#_Toc474167174)

[5 Communication with other microservices 8](#_Toc474167175)

[5.1 Communications with the Dataset MS 8](#_Toc474167176)

[5.2 Communications with the Study Card MS 8](#_Toc474167177)

[6 Open questions 9](#_Toc474167178)

[6.1 Question MK: How many Nifti files is there and what is the average size of each file? 9](#_Toc474167179)

# Microservice context

## Microservice presentation

This microservice is a part of the Shanoir-NG application and is responsible for converting dicom files to nifti format.

## Main functionalities

Shanoir NG Dcm2Nii main functionalities are:

* Managing Nifti converters (create, edit but not delete)
* Converting dicom files to nifti format with a chosen converter

## Application users

Target population is:

* Shanoir users of any level
* Shanoir administrators for nifti converters management

# Database architecture

## Nifti Converter

### Name

Mandatory

### Is Active

Boolean, Mandatory

### Ref Nifti Converter Type Id

Mandatory.

Link to the Ref Nifti Converter Type table.

### Comment

Free text.

## Ref Nifti Converter Type

### Label Name

Mandatory.

The converter type could be one of these: dcm2nii, mcverter or clidcm (for the old versions).

## Study Card

Link to the Study Card table with the nifti converter’s id. A converter is linked to a study card to keep the homogeneity of the data, so that for the same center and the same acquisition machine, nifti files are converted with the same converter.

## Dataset Expression

Link to the Dataset Expression Table with the nifti converter’s id, name and the isOriginalNiftiConversion Boolean.

# Front-end architecture

## Manage Nifti Converters

This page is only accessible for the administrators.

They can add or edit nifti converters here but not delete them.

All nifti converters, active or not, is listed in a table, with the name, type, isActive, comment column. At the right of each converter, a button “Edit” allows to update a converter in another page.

# Back-end architecture

## 4 Scenarios for nifti conversion:

The nifti conversion could be triggered in one of these scenario:

### Import

An conversion is done during the import process between the anonymization and the study card application steps. The output folder could be found with the “show the details” link, for example, “/data/shanoir-dev\_nifti/subject\_2/examination3/examination”.

During the import, the Dicom files are converted to nifti using the converter defined with the study card chosen at the beginning.

### View Dataset Detail

Sometimes, the nifti conversion may have failed during the import, or a user wants to use a dataset with a different converter. It is possible to (re)generate nifti files for the dataset in its detail page. Under the “Available Nifti files to download for this dataset” is the list of already converted nifti files (At least the nifti files generated during import). It gives the converter and its version used to obtain those files.

To regenerate files, a window let user choose one converter at first. If the corresponding nifti files do not already exist, the conversion begin; otherwise, a message “Nifti files already exist for dataset xxx using xxx converter.” is shown. At the end of the conversion, there is a new link under the Available Nifti files and the user is informed by a message “Nifti files have been generated for dataset xxx with xxx”.

### Manage Datasets

Users can choose one or more datasets to be converted from the manage datasets table. The conversion part is the same with the View Dataset Detail page.

### View Detail Study tree

Users can choose one dataset to be converted from the view detail study tree. The conversion part is the same with the View Dataset Detail page.

## The conversion steps in details

Whatever the scenario which triggers the nifti conversion, the ***convertToNiftiExec*** method is always called to do the conversion. For the import process, the method is called by the ***convertToNifti*** function; otherwise, it is called by the ***reGenerateNiftiFiles*** function.

Below a schema to explain the conversions steps in details:

(re)GenerateNiftiFiles

ConvertToNifti

convertToNiftiExec

dicom2niftiExec

dcm2niiExec

clidcmExec

mcverterExec

Nifti output temp folder

Nifti destination folder

Remove temp folder(s)

Message to user

Update Dataset MS with Nifti files

### For the dicom import process:

The imported dicom files could come from either a Dicom zip file or from network or preArchieve. If they are from a zip file, the images are copied in a temp folder like “C:\Users\yyao\AppData\Local\Temp\folder\_5fb8b31e-15a03148644--7fe99097460118931060578.upload\SERIES” during the import process. Otherwise, they could be accessed by a folder path directly. In any situation, the input dicom files could be presented by a folder path, which we called “***tempFolderPath***” later.

In the tempFolderPath, we create firstly a folder with the id of the serie, like “C:\Users\yyao\AppData\Local\Temp\folder\_5fb8b31e-15a03148644--7fe99097460118931060578.upload\SERIES\1.3.12.2.1107.5.2.36.40296.2010030419031359358071443.0.0.0.”

Next we create directories and put images inside. 1 directory = 1 dataset. For example, \dataset0 contents IM000545, \dataset1 contents IM000544 and \dataset2 contents IM000546, IM000547 and IM000548. Let’s call these directories “***datasetDirectories***”.

***datasetDirectories***

When the ***datasetDirectories*** folder is prepared, we iterate over the whole folder to do the conversion of all the dicom images using the convertToNiftiExec method which is presented in the 4.2.3 paragraph.

Note that we don’t need the conversion if there are spectroscopy files.

### To (re)generate nifti files for dataset(s):

At first, we call a function to check if the chosen dataset(s) is regenerable or not. If not, we stop here and show a message immediately.

This time the dicom files are not directly accessible on the user’s host. They will be downloaded from the PACS and set in a temporary folder at first.

As the 4.2.1 part, the ***tempFolderPath*** begins with something like ““C:\Users\yyao\AppData\Local\Temp\”. After that we add a temp folder name calculated by the dataset expression (Dicom), the subject name and the dataset id. So the tempFolderPath looks like “C:\Users\yyao\AppData\Local\Temp\tmp-DICOM.patient1.datasetFilesToConvert.id.158”.

To get the ***datasetDirectories*** then, we consider one dataset file is an url to the PACS server, like <http://localhost:8080/wado?requestType=WADO&studyUID=1.3.12.2.1107.5.2.36.40296.30000010030206191446800000091&seriesUID=1.3.12.2.1107.5.2.36.40296.201003041909249627171814.0.0.0&objectUID=1.3.12.2.1107.5.2.36.40296.201003041909247578871810&contentType=application/dicom>.

All the dicom files are downloaded with the url, into their ***datasetDirectories***.

We iterate over the whole ***datasetDirectories*** folder to do the conversion using the converToNiftiExec method presented in the 4.2.3 part.

Note that after the conversion of one dataset, the Dataset MS need to be informed with the updated nifti files. So the id of the dataset is required to be saved. We put the dataset id in the name of the temp output nifti folder, ***tempOutputFolder*** (e.g. “C:\Users\yyao\AppData\Local\Temp\tmp-nifti-24-158”).

Note also that sometimes the subject name could be changed after the import, or some problems occurred during the import, so before the conversion we need to check if there are files that exist already in the ***tempOutputFolder*** and if each folder for constructing nifti folder exists already.

### The convertToNiftiExec function

The conversion is a command line to be executed. To construct this command line, we do a quick check on the converter to use. A command line is called according to the converter’s type used.

|  |  |  |  |
| --- | --- | --- | --- |
| Converter used | Function to call | Is 4D | Comment |
| mcverter | mcverterExec | yes |  |
| clidcm | clidcmExec |  | Load the clidcm converter because the converter can switch from dcm2nii to clidcm in case of 4D volume sequence. Search for a ‘.pop’ file. If found, then create a ‘.bvec’ and a ‘.bval’ file from the ‘.prop’ file. |
| Dicom2Nifti | dicom2niftiExec | yes |  |
| other | dcm2niiExec |  |  |

The convertToNiftiExec function needs some parameters as the converter used, the input folder, the output folder (same as the inputFolder, which is also the datasetDirectories), the is4D Boolean and the isConvertClidcm Boolean.

### The post-conversion

After the conversion, the already generated nifti files are moved from the datasetDirectories into a temp output folder at first.

They are then moved to the final destination folder. The destination folder is calculated by the “nifti.destination.folder” attribute configured in the “Shanoir.properties” file, the modality folder (like “/subject\_2/examination3/examination”) and the nifti converter’s name (e.g. “dcm2nii\_2014-08-04”).

Remove all the files created during the conversion process, like the datasetDirectories folder, the temp output folder and the temp folder containing retrieved dicom files from the server if they are not from a dicom import.

Check in the final destination folder if one of the output files is a “.prop” file. If it is the case, show an error message to let user know that there has been an error. If not, tell user that the conversion is fine with a success message.

If it is not an import process, the Dataset MS needs to be informed that for the dataset has been updated with the new nifti files. The dataset expression , the dataset file will be updated at the same time.

## The conversion black box

To resume the details shown before, the nifti conversion microservice can be considered as a black box. The parameters of entry could be the converter, the folder (or the list of folders) of the dataset(s) to be converted and the file path of the nifti output folder.

Only the dataset(s) folder(s) is mandatory. The converted nifti files are saved in a destination folder defined in “Shanoir.properties” file if not specified. In case there is no converter selected in the study card, the converter is loaded according to the default nifti converter type (dcm2nii or mcverter) specified in the “Shanoir.properties” file and the most recent version of this type is used.

Nifti converter

Dataset(s) folder path

Nifti Conversion

Converted Nifti file

Final destination folder path

# Communication with other microservices

Nifti conversion microservice communicates with the Dataset MS, the Study card MS.

## Communications with the Dataset MS

To get the dataset(s) from its id, we need to interact with the Datast MS.

Once the nifti file(s) generated for the dataset(s), the Dataset MS needs to be informed that the dataset(s) has been updated with the new converted nifti files. The dataset expression and the dataset file will be updated at the same time.

## Communications with the Study Card MS

To get the converter defined for one study card, we need to interact with the Study Card MS.

# Open questions

## Question MK: How many Nifti files is there and what is the average size of each file?

Answer YC: Until December 2016, there are 205 964 nifti files who take 2,1 To. So the average size of one nifti file is about 10 Mo.