# Leveraging Open source/Freeware Tools and Design Methodologies in Translational Research: A Siemens-based Case Study -- from Raw Diffusion and Structural Data to PACS and the Web

R. Pienaar[1], R. Wang[1], B. Fischl[1], V. Wedeen[1], and P. E. Grant[1]

[1]Radiology, Massachusetts General Hospital, Boston, MA, United States

INTRODUCTION: Most modern MRI scanners are typically controlled from an operator console that is itself a specialized personal computer -- often running Windows or a custom Linux distribution. These scanner consoles can be easily interconnected to broader hospital processing networks, often times in ways that the original manufacturer did not anticipate. In the simplest sense, such interconnectivity can result in faster data delivery between different domains in the hospital or even remote administration and deployment of scanner protocols. Moreover, by leveraging open source tools and frameworks, it is possible to connect the scanner directly to advanced image post-processing pipelines, resulting in sophisticated and rich delivery of information across both research and clinical domains. This paper presents a case study in which open source / freeware tools and design methodologies are used to connect a standard Siemens MRI system to two common postprocessing cases: (i) a DTI/DSI tractography study based on the freeware package 'TrackVis' and (ii) a surface reconstruction based on the freeware package 'FreeSurfer'. In both cases, resultant outputs are formatted for easy access and display (such as in a web browser) or repackaged and transmitted to a PACS. The more general underlying design approaches and methodologies are also discussed in detail. By providing value-added information in an easily accessible manner, such pipelines can ultimately aid in decisions affecting clinical care and therapy.

METHODS AND RESULTS: The figure below shows a hardware topological map connecting various components together as well as an automated translational pipeline. The hardware consists of: a Siemens MRI Scanner with associated operator host console (OHC), MR-Image Reconstruction (MRIR) unit, external hospital PACS, and a postprocessing Linux workstation (PPLW). The PPLW houses the main software pipelines connecting the scanner DICOM data to the Hospital PACS or more general web services. Several processing pipelines exist in the PPLW with each processing pipeline consisting of a serial chain of analysis sub-units (ASUs). The Main Processing ASU contains two sub-pipelines corresponding to diffusion and reconstruction processing. Depending on actual physical location and need, the PPLW can be directly connected to the local scanner LAN, or (more typically) be located anywhere in the broader hospital intranet. In the case of direct connection to the Scanner LAN, the PPLW can access scanner data directly using the open source 'samba' package in the case of Windows-based scanner consoles, or natively using the open source network file system (NFS) for Linux based consoles. Furthermore, the PPLW can be configured to receive PACS transmission of scanner data using the open source 'dcmtk' toolkit.
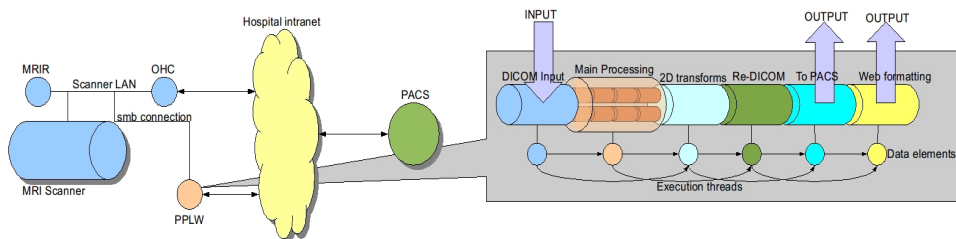
For the processing pipelines, we exploit the fact that data and processing are fundamentally serial in nature. Each ASU has a well defined set of pre-conditions, and in turn presents a well defined set of post-conditions, which typically become the pre-conditions to subsequent stages. These pre- and post-conditions are often data elements and constructs, and can be easily asserted independently of the processing pipeline. This relatively straightforward pattern of I/O means that the pipelines themselves can be reliably constructed using standard UNIX shell programming (i.e. the bash shell). Threads of processing can weave through stages in a fundamentally serial manner, skipping stages as needs pending internal states or user specified constraints. Each ASU typically consists of an independent image processing tool. The processing pipeline is glue technology that robustly connects these components together.

Common Pipeline Source Inputs: There are two basic methods of input -- in the first case (specific to a Siemens scanning backend), the PPLW accesses raw scanner directly off the OHC or MRIR; in the second case (non-Siemens specific), studies of interest are transmitted to the PPLW from a standard hospital PACS. Direct access to the scanner relies on network technologies such as 'samba' or 'NFS', and allows for a high degree of automation. The PPLW can routinely examine scanner directories, and on finding scanned data, independently initiate a processing pipeline. A more general method of access is to simply run a PACS listener on the PPLW such as 'dcmtk'. This package will receive PACS transmissions, saving DICOM data locally. This input method is less automated than direct access since the PPLW needs an asynchronous event (the DICOM transmission) to initiate processing.

Main Processing Case Study: Siemens DTI/DSI to web and PACS: Diffusion DICOM data (either DTI or DSI) are automatically detected by the PPLW (either by monitoring the scanner or after receipt from a PACS transmission). Images are then converted to NIfTI format, automatically eddy-current corrected (using the freeware FSL toolset), and then processed by the freeware package 'TrackVis'. A 3D tract volume is displayed by 'TrackVis' and systematically sliced into a series of flat images along each cardinal plane (sagittal, axial, coronal).

Main Processing Case Study: Siemens MPRAGE to web and PACS: DICOM inputs are processed directly by the freeware package 'FreeSurfer', ultimately resulting in reconstructed surfaces and output text file reports. Using native 'FreeSurfer' tools, the surface is rendered in an automatic fashion and off-line 2D image snapshots made. Similarly, segmented volume data is rendered and 2D image snapshots made. All rendering and image capture is performed without any user intervention.

Common Pipeline outputs: Regardless of the main internal processing of a pipeline, in most cases there are common final outputs: text reports, and flat images. These outputs can be formatted for display in dynamically generated web pages, or alternatively images in particular can be re-converted back to DICOM format and transmitted back to a PACS system for clinical access and display.

CONCLUSION: The fully automatic processing pipelines discussed here are constructed using only open source/freeware tools, access DICOM data as collected on a Siemens scanner, perform sophisticated post processing, and generate output to the web or a hospital PACS – all for the cost of a Linux workstation. Any processing scheme that is of primarily serial nature and command-line controlled is amenable to this automated framework. Such a pipeline framework can provide a high value added proposition with little hardware and development/deployment costs.