MeCS – Integrating prototype processing programs into clinical routine

B. W. LEHR[1], F. SCHWESER[1], A. DEISTUNG[1], D. GÜLLMAR[1], AND J. R. REICHENBACH[1]
[1]MEDICAL PHYSICS GROUP, DEPARTMENT OF DIAGNOSTIC AND INTERVENTIONAL RADIOLOGY I, JENA UNIVERSITY HOSPITAL, JENA, GERMANY

INTRODUCTION – MRI data processing in daily clinical routine is usually conducted directly at the MRI workstation. However, newly developed processing methods are often not available at the MRI workstation until several years after their original development. Furthermore, available methods suffer from limited functional range, on the one hand for licensing reasons but on the other hand due to the limited computational power of the MRI workstations. Yet, new processing methods are available, as prototype processing programs (PPs). These PPs are developed during research, designed in optimized numerical calculating environments (MATLAB [1], IDL [2]), employing purpose-built packages (FSL [3], Freesurfer [4]), or require powerful hardware. If such PPs were available for clinical routine, improvements could be applied during method development, studies could be conducted earlier, and finally powerful diagnostic tools could be made widely available that were only accessible by a small number of selected institutions before. However, efficient usage of PPs for clinical routine depends on tight integration into the existing workflow as well as on high usability. Therefore, this contribution presents the Medical Computation Server (MeCS), to our knowledge the first approach to translate prototype processing programs into clinical routine.

PRINCIPLES OF OPERATION – Fig. 1 schematically illustrates the tight integration of MeCS into the clinical workflow. MeCS' DICOM [5] node is connected to the clinical data network, like imaging system nodes (e.g., MRI, CT), examination workstations, or picture archiving and communicating systems (PACS).
User guidelines: Currently, measured and processed MR images are transferred directly to an archiving node (PACS) (left and middle threads in Fig. 1). Using MeCS, unprocessed DICOM images are sent to the DICOM node associated to the MeCS server (right thread in Fig. 1). Subsequently, a processing order has to be assigned. For this, the transmitted dataset, the desired PP, the target node for the result (e.g., PACS, workstation), and (if applicable) certain processing parameters are selected via a web application. Subsequently, MeCS initiates computation of the data by the selected PP on a remote high performance computer and transfers the processed DICOM images to the specified target DICOM node.
PP Developer Guidelines: A PP to be integrated into MeCS requires DICOM file format read and write capabilities. This ability can be added to practically every PP by wrapping it in a script (e.g. bash [6], python [7]) overtaking data conversion to and from more suitable formats. Furthermore, a configuration file must be provided for each PP, specifying the execution command, an optional list of parameters to query, and the type of supported data (e.g., phase or magnitude data, diffusion tensor imaging, or perfusion weighted imaging sequence data).

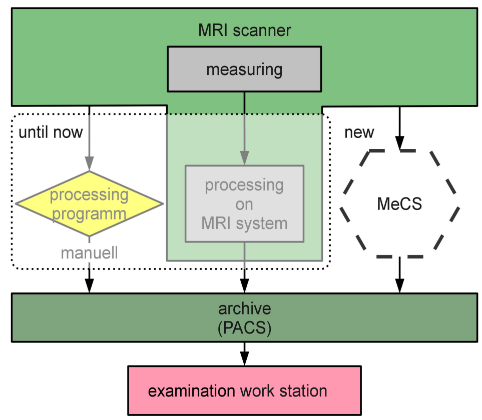

FIG. 1. Schematic of the integration of MeCS in the clinical workflow.

DESIGN OF THE SERVERS – Fig. 2 illustrates the data flow between the three modules of MeCS: the *DICOM Server Module*, the *Assigning Module,* and the *Processing Module*.
DICOM Server Module: This module represents the DICOM node of the MeCS that stores the incoming data. It extracts meta-information (e.g., TE/TR, slice thickness, sequence name) from incoming DICOM files and provides them to the *Assigning Module*.
Assigning Module: This module operates the web application used to assign processing orders. It is accessible from any computer within the clinical network but restricted to registered users. The information collected by the *Assigning Module* about a processing order is saved in a "status file".
Processing Module: This module creates a home folder for each processing order. Inside the folder it provides a predefined location containing the selected input data as selected in the processing order provided by the *DICOM Server Module*. It also provides the runtime parameters for the PP in a "track sheet" file in this folder. The *Processing Module* executes the PP on a high performance cluster (HPC). After successful processing, the module adds a comment to each DICOM file to indicate that the data was processed by MeCS. Finally, the module initiates the data transfer to the designated target node.

RESULTS – MeCS was integrated into the network of the Jena University Hospital. The *DICOM Server Module* is built on a CTN [8] Server with extended metadata extraction using CTN's standard MySQL database [9]. The *Assigning Module* is realized as PHP [10] application, running on an Apache [11] server, authenticating against a LDAP [12] user database. The "status file", as well as the "track sheet" store information in yaml [13] format. Finally the *Processing Module* is also implemented as PHP program. Currently, PPs are available for susceptibility weighted MR imaging [14], quantitative susceptibility mapping [15], and diffusion tensor imaging [16]. MeCS executes PPs remotely on a HPC cluster (96 Cores, 192 GB RAM), scheduled by a Sun Grid Engine (SGE) [17]. The usage of the SGE facilitates application of parallelized algorithms and takes care of efficient load balancing for simultaneous work on multiple processing orders.

DISCUSSION – MeCS allows the integration of prototype processing programs in a clinical environment, providing equal usability compared to processing methods available on the MRI workstation. Thus, MeCS is supposed to improve efficient method development and decrease the time gap between initial development of a novel processing technique and final review studies for elaborating clinical benefits. Enhancements can easily be added due to the modular structure, an alternative assigning module could for instance automatically select a PP and default parameters depending on the data type. Further improvements will include the ability to process non-DICOM data like spectroscopic measurements or return vector objects like fiber tracts.

LITERATURE: – [1] Mathworks, Natick, MA, USA [2] ITT Visual Information Solutions, Boulder, CO, USA [3] FMRIB, Oxford, England [4] Athinoula A. Martinos Center for Biomedical Imaging, Massachusetts General Hospital, Charlestown, MA, USA [5] National Electrical Manufacturers Association, Rosslyn, VI, USA. DICOM, 2009. ISO Standard 12052 [6] Chat Ramey et al., http://tiswww.case.edu/php/chet/bash/bashtop.html [7] Python Software Foundation http://www.python.org [8] CTN DICOM Server; http://erl.wustl.edu/research/dicom/ctn.html [9] Oracle Corporation, Redwood Shores, CA, USA [10] The PHP Group, www.php.net [11] Apache Software Foundation, Forest Hill, MD, USA [12] The Open Group. Lightweight directory access protocol (ldap) [13] Clark Evans et al, http://www.yaml.org [14] Reichenbach J R et al. NMR Biomed, **2001**, 14, 453-467 [15] Schweser F et al. NeuroImage **2010** (in press) [16] Pierpaoli et al. Radiology, Radiological Society of North America, **1996**, 201, 637 [17] Sun Microsystems, Santa Clara, CA, USA

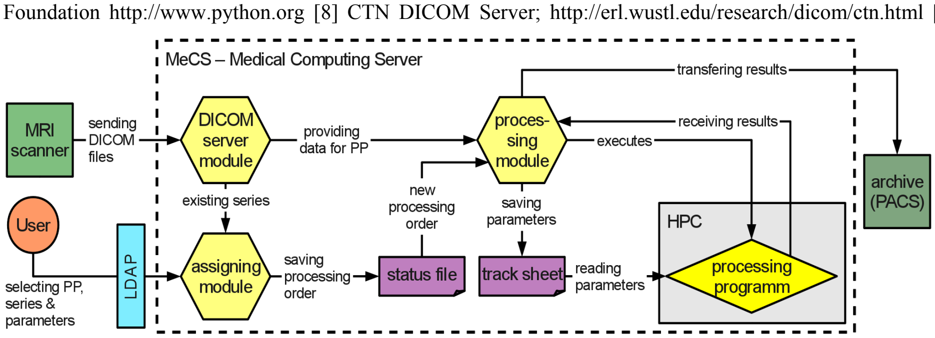

FIG. 2. Schematic of data flow inside the MeCS. Yellow hexagons represent the modules of MeCS, purple boxes files, and green boxes denote DICOM nodes.