## A framework for programming a modern magnetic resonance system

## D.C. Pizetta, L.C.R. Silva, E.L.G. Vidoto, M. Martins, A. Tannús

São Carlos Institute of Physics, University of São Paulo, São Carlos, Brazil e-mail: daniel.pizetta@usp.br

**Introduction:** Magnetic resonance system leads the tools of analysis and diagnosis because its non-invasive and non-destructive characteristics over the study object. Recent advances in the magnetic resonance field introduced such technology in more accessible and portable design which requires flexibility for being used to new applications that researchers could create by their own. This flexibility and liberty is not found in commercial system. The aim of this project is the definition of a framework that assists the equipment programming and management, including since the creation of the interface to data processing.

**Methods:** The entire framework is supported by up-to-date programming techniques that generates an API that can be integrated with others external tools such as simulators and scientific libraries. The main language is Python because of its simplicity and availability of many scientific libraries such as NumPy, SciPy, etc.

**Results:** Within this framework, the PyMR [1] framework offers a generalized application programming interface (API) for programming and setting magnetic resonance systems, while another derived API constitutes the backend, which communicates in a lower level with the hardware. Preliminary versions are already in use in local magnetic resonance system at University of São Paulo, São Carlos Institute of Physics, Center of Magnetic Resonance Imaging and Spectroscopy. Figure 1 shows the interface of <u>object</u> manager which has the object creator, object inspector and object editor. The first one shows all the classes that can be instantiate. The second one shows the system structure and the last one shows the properties that can be changed for each part of the system. Plugins were also produced to help users to manage to create, in any level, new applications or systems. These plugins should be integrated with the Eclipse Integrated Development Environment (IDE) or other ones. The whole framework and plugins have an integrated documentation which could be exported in HTML or PDF formats.

			Object Manager				
bject Creator	5 8	Object Inspector		8 X	Object Editor		
View		Object			Filter		
Class Hierarchy		<ul> <li>Image: Image: Ima</li></ul>			pymr.core.peripheral.amplifier.RFAmplifier : amp_rf_tomce		
Filter		Filter			Parameter	Value	
Class	Documentation	Object ^	Class	Label	PyMRObject	Folde	
pym.core.device AnalogDigitalConverter CascadeIntegratorCombFilter CascadeIntegratorCombFilter Device DigitalAnalogConverte EmphasisFilter Filter Filter Filter Filter InitelmpulseResponseFilter NumericallyControlledOscillator Transducer	Analog-to-digital co Cascade integrator- Class that implemer Digital-to-analog co Emphasis filter class Filter class. Filter class. Filter class. Finite impulse resp Infinite impulse resp Infinite impulse resp Sumerically Control Base class for trans Class that represent	<pre>F mrs_hahn F mrs_hahn transmitter_set ttl ttl_roc_pm ttl_proc_pm ttl_proc_m ttl_proc_am ttl_nco_secondary ttl_nco_primary</pre>	System TransmitterSet Transmitter Processor Processor Processor NumericallyControlledOscillator DigitalAnalogConverter Prof RFColl Prof RFC	MRS Hahn Spect           Tx Set           Tx 1           PM Processor           AM Processor           AM Processor           AM Processor           TO Secondary           NCO Finany           DAC           TOMCO BT00500 SET           TOMCO BT00500           TOMCO BT00500           TOMCO BT00500	description enabled label TOMCC model BT0051 name amp_rf serialNumber 7.76e+ vendor 7.76e+ Vendor 7.76e+ Peripheral Amplifier BFAmplifier amplificationBandwidth Broadb amplificationClass A amplificationDevice Tube amplificationDevice Tube	TOMCO BT00500 BT00500-Gamma amp_rf_tomco_1 7.76e+03 TOMCO Broadband A Tube ContinuousWave	Jamma meo_1  PyMR Installation Notes to developers Authors  PTutorial
GradientSet	Class that represent	- amp_rr_tomco_2	RFAmplifier	TOMCO BT00500	fallTime frequency	23 MHz	Parenting: the object hierarchy
PhaseGradient ReadGradient SelectionGradient _PreEmphasis Receiver ReceiverSet	Class that implemer Class that implemer Class that implemer Class that implemer Class that implemer Class that implemer	receiver_set rx1 rx1_fir_secondary rx1_fir_primary rx1_fir_primary	CREceiverSet CREceiver Casceiver FinitelimpulseResponseFilter CascadeIntegratorCombFilter AnalogDigitalConverter CREPreamplifier CREPreamplif	Rx Set Rx 1 FIR Secondary FIR Primary CIC ADC Miteq AU-1448 Magnet 0.5T NdFeB	gainRange [ <quantity(10, 'de<br="">gateDelay 1 gateLogic Lu gateType B inputMpedance 55 inputPowerRange [<quantity(1, 'mil<="" td=""><td>10 dB 10, 'decibyte')&gt;, <quantit 1 us Low Blank 50 Ω</quantit </td><td>Bibliography Changelog</td></quantity(1,></quantity(10,>	10 dB 10, 'decibyte')>, <quantit 1 us Low Blank 50 Ω</quantit 	Bibliography Changelog

Figure 1. Some characteristics of PyMR framework. a) Object manager formed by object creator, object manager, object editor. b) Integrated documentation generated in HTML.

**Conclusion:** The set of frameworks and plug-ins is extremely versatile and flexible, reflecting the high power of modern technologies or new applications for existing ones.

**Referências:** [1] PIZETTA, D. C. Biblioteca, API e IDE para o desenvolvimento de projetos de metodologias de Ressonância Magnética. 2014. Dissertação (Mestrado em Física Aplicada) - Instituto de Física de São Carlos, Universidade de São Paulo, São Carlos, 2014. Disponível em: <a href="http://www.teses.usp.br/teses/disponiveis/76/76132/tde-28042014-160738/">http://www.teses.usp.br/teses/disponiveis/76/76132/tde-28042014-160738</a>