

G-PST/ESIG Webinar: Open Source Tools for System Operators

Question	Answer
You identify need for code review, version control and unit testing. What is in place for power-systems open source? Who supports this and is it sustainable?	Those should be part of the regular development done by the development team. As alluded in the presentation it can be valuable if some of the users with high stakes in the security aspects participate in the code review - having an extra pair of eyes from the outside should improve trust and security further. Version control and unit testing should be part of good programming practice and not seen as an extra cost. Code review as well, but at the level required by critical applications, it can require an extra effort. It needs to be baked into the resources the community partners have. Open-source projects tend to be flexible - if there are less funds and less hands, things slow down, new features are implemented slower, but this does not prevent from prioritizing critical tasks like code review.
Are you interested in doing a comparison among different tools for capacity planning, some of which may have capabilities that go beyond the commercial tools?	Yes, this has been the intention of Pillar 5. The ball has been kicked forward as the previous benchmarks took more effort than expected. You can learn more by joining the Pillar 5 working group.
Was fault analysis carried out in your benchmarking exercise? If yes, how did the results compare from different platforms?	None, we recognized that this would be a nice next step. But for simplicity, we did not do any fault analysis.
Is there research comparing open source power flow tools to commercially available tools?	There are some comparisons, but I'm not aware of much public material. I know that RTE put significant effort to benchmark/compare PowSyBL and some commercial tools. Also PowerModels.jl has been validated against PSS/E for a limited problem specification. Pandapower too... It was our recognition that most open-source power systems projects do some benchmarking against commercial tools if there are existing capabilities. The commercial tools have errors and problems too. So part of the benchmarking idea is to provide an agreed upon solution so that anyone can make comparisons and point out issues. Ideally we could extend this to many other types of problems.
How important is it to utilities to have a downloadable app with a GUI rather than a command-line app that installs via a package manager?	This is speculation as I don't have direct information, but I believe it depends on the setup. If the utility has systems where it can hook-up command-line apps into their workflows, then it may not be a major issue. But if the tool is serving purposes outside existing workflows or if those tools are not easily configurable to include command-line apps, then GUI can be valuable. A GUI can be valuable in any case for testing and comparison purposes before any decisions on adoption are taken.

<p>Are there any power flow tools that can open a .raw file and be used to analyze real systems?</p>	<p>PowerModels.jl supports PSS/e-v33 .raw files. PowerSystems.jl supports the same, with new capabilities to read other versions as well. The PowerModels.jl developers also created a Python parser for raw files. PowSyBI has raw file parsing capabilities as well, but I'm unsure of the details.</p>
<p>Julia does not yet provide needed capabilities for complex real-world models (e.g., explore generated MP and MP results)? What development is planned?</p>	<p>I'm not sure that I'm follow with the question. If MP is "mathematical programming", then I'd push back and claim that the JuMP.jl package is quite mature and provides an emmense set of capabilities with a very nice layer of abstraction that makes it easy to build large-scale models. JuMP is a great example of successful open-source projects that has implemented many of the requirements that Juha outlined (not power systems specific).</p>
<p>Juha, what can we expect from the new Mopo opensource software ? On what software will it be developed ?</p>	<p>It will continue the development of Spine Toolbox (data, scenario and workflow management for modelling) and SpineOpt.jl (energy system planning and operations with physics and temporal adaptability). See https://github.com/Spine-project.</p>
<p>Have you done any analysis on CIP compliance with the use of open source tools?</p>	<p>I have not, but it seems like it should be possible and a great use of open-source tools.</p>
<p>What do the panelist think about openmodelica as powersystem modeling software?</p>	<p>OpenModelica has some great features for waveform analysis and algebraic modeling. We explored using it for some of our dynamic simulation tool development. Ultimately, we opted to develop our tools in a more general language (Julia) for access to analytic, optimization, and other general software features. I believe Dynawo uses open modelica</p>
<p>Can you give some examples of utilities or system operators that are using open source tools today</p>	<p>A few specific examples are RTE, EnergiNet, and Alliander</p>