

State of the Art and Future of OSS for Power Systems

F. Milano, *Member, IEEE* and L. Vanfretti, *Member, IEEE*

Abstract—This paper presents the main concepts of Free and Open Source Software (FOSS) that are relevant for power system analysis, research and education. The main FOSS projects for power system analysis are briefly introduced and discussed. The paper also provides some outlines about the future of FOSS for power systems and the activities of the IEEE Task force on Open Source Software.

Index Terms—Open source, software, simulation, power system education.

I. INTRODUCTION

A. Motivation

MOST power system software packages are commercial proprietary products that require a generally costly license. This fact is implicitly accepted as normal in the power system community. However, there can be a reliable and costless alternative. This alternative is provided by Free and Open Source Software (FOSS). In this paper, we show that FOSS is a valid platform to distribute educational, research and industry-oriented tools for power system analysis as it has proved to be in several other scientific fields.

B. Literature Review

After several technical frustrations imposed by proprietary software in 1983 Richard Stallman [1] responded with a plan to develop a complete system built from free software. He named the system GNU (as a recursive acronym “GNU’s Not Unix”) and later in 1985 founded the Free Software Foundation (FSF) [2] to not only disseminate Free Software, but more broadly, the ideals of freedom and cooperation that were present at MIT’s AI lab at those days. Later in 1991, Linus Torvalds released to an FTP site the Linux kernel under the GNU GPL [3] which by 1992 resulted as the first free operating system by combining it with the GNU operating system [1]. In 1998 a group of programmers from the FSF movement decided to launch the Open Source Initiative (OSI) [4] with the purpose of gaining greater corporate acceptance for the free software development model. Claiming that the term “free software” was too confusing for corporate business and choosing the term “open source” they aimed to increase

the attraction towards the free software development model from investors and mainstream business.

A detailed discussion of the history of FOSS is out of the scope of this paper. However, we provide references for interested readers. In particular, current perspectives on FOSS are described in [5], and recent issues on OSS are extensively discussed in [6] and [7]. Those interested in recent government policy in the U.S. towards open source software are referred to [8]. A comprehensive analysis of FOSS licensing details is discussed in [9]. Readers looking to take benefit of FOSS in their power business are referred to [10] and [11]. Finally, a very compelling publication deals with the philosophical implications of FOSS [12].

Remarkable efforts have been made by the academic community to provide educational software for teaching power system topics. However, most available tools do not fully apply the FOSS rules and present at least one of the following deficiencies: a lack of wide distribution of the software, access to the source code of the software, a community of users, a comprehensive documentation, and the distribution of the software with some kind of license that allows freely modifying and redistributing the code (e.g. the GPL [3]).

In this paper, we only focus on educational and/or research software packages that are distributed for free and whose code is open. These are: UWPFLOW [13], TEFTS [14], MatPower [15], VST [16], PSAT [17], InterPSS [18], AMES [19], DCOPFJ [20], Pylon [21], and OpenDSS [22]. The order is chronological and based on the first public release of the software package.

C. Contributions

In summary, these are the novel contributions of the paper.

- 1) To present the main concepts of FOSS that are relevant for power system analysis, research and education.
- 2) To provide an overview of the most important existing FOSS projects for power system analysis.
- 3) Discussing the future of FOSS for power systems, including the activities of the IEEE Task force on Open Source Software [23].

D. Paper Organization

The remainder of this paper is organized as follows. Section II describes common concepts of FOSS. Section III presents the main FOSS projects for power system analysis. Section IV discusses the activities of the IEEE Task force on Open Source Software, while Section V provides remarks on the future of FOSS for power system analysis. Finally, in Section VI, conclusions are duly drawn.

Paper submitted to 2009 IEEE PES General Meeting.

F. Milano is with the Department of Electrical Engineering, University of Castilla-La Mancha, 13071 Ciudad Real, Spain. E-mail: Federico.Milano@uclm.es.

L. Vanfretti is with the Department of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY, USA. E-mail: vanfri@rpi.edu

F. Milano is partly supported by the Ministry of Science and Education of Spain through CICYT Project DPI2006-08001; and by Junta de Comunidades de Castilla-La Mancha, through project PCI-08-0102.

II. BACKGROUND

Software can be divided into three categories based on the development method: Free Software (FS), Open Source Software (OSS), and Proprietary Software. Free and Open Source Software (FOSS), as we will explain, is a term used to merge common characteristics of Free Software and Open Source Software. In this Section, we (i) introduce concepts and definitions of each type of software development and licensing; (ii) describe areas of the power business where FOSS software can be adopted; and (iii) briefly discuss how the different areas of the power systems community can take advantage of FOSS.

A. Concepts and Definitions

1) *Proprietary Software*: is the term used to refer to software that has restrictions for its use, modification and, more importantly, restrictions on copying, distributing, and publishing unmodified or modified versions of it. The restrictions are placed by the proprietors of the software and are detailed in the software license. In the U.S., copyright laws provide severe penalties for unlawful distribution of copyrighted material. Reverse engineering of the software could also violate the U.S. Copyright Law or the Digital Millennium Copyright Act [24].

Proprietary software is also referred as commercial, non-open, or non-free software. Sometimes these terms are considered derogatory, which is actually unintended, due to the fact that terms like *freedom* and *openness* are more appealing than their opposites. In [25], the term *closed software* has been coined to refer to all this type of software and avoid the derogatory misunderstanding.

2) *Open Source Software*: is one that complies with the Open Source Definition (OSD) [26], published by the Open Source Initiative (OSI) [4]. Formed in 1998, OSI is a non-profit corporation created by a group of programmers to promote the adoption of OSS licenses. The OSD gives the criteria to which software wishing to adopt an OSS license must comply with. These criteria are that users must be free to use the software for any purpose, make copies and distribute the software without paying to the issuer of the license, to create derived works and to distribute them without paying royalties, to view and use the source code and to use the open source software in combination with other software *including proprietary software*.

In summary, open source software permits anyone, anywhere and for any purpose to copy, modify, and distribute the software for free or for a fee. Therefore, anyone has to have full access to the source code.

3) *Free Software (FS)*: is software that can be used, studied, and modified without restriction. The term *free software* was coined by Richard Stallman who founded the Free Software Foundation, its formally defined by the Free Software Definition [2]. One of the most important instances of the definition is that software can be copied and distributed in modified or unmodified form without restrictions. Restrictions may be used only to ensure that future recipients of the software are guaranteed to copy, study, modify and distribute the software (this is the main difference with respect to OSS). Moreover,

the source code of the software must be made available, and it may be accompanied by a software license. The license should state that the copyright holder permits these acts, or alternatively the software can be released into public domain so that the rights mentioned above automatically hold. The Free Software Foundation maintains a list of Free Software Licenses [27], being the most common the GNU General Public License (GPL) [3].

It is customary to explain the FS concept by saying that the idea is not that the software should be free “as in beer”, or available at no charge, but that it should be free “as in speech”, so that the software can be reviewed and modified.

4) *Free and Open Source Software (FOSS)*: FOSS, also known as FLOSS (Free/Libre Open Source Software), is a merger of the FS and OS concepts focusing on the characteristics for software development and distribution without addressing subtle differences between the two of them. To this extent, Free Software and Open Source Software are near synonyms. A clear difference between the two is explained in subsection II-C, which describes software licenses. Interested readers can find details on the philosophical background and differences between FS and OSS in [1], [9], [12], [28].

5) *Successful FOSS Projects*: The list of successful FOSS projects is impressive. For the sake of illustration, we cite the Apache Project [29], and the Mozilla Project [30]. The Apache Software Foundation [29] is the developer of the Apache Web server that “*is the most widely deployed server with 70 percent of the 54 million Web sites queried in the Netcraft data collection*” (study by Netcraft) [5]. Apache shows how the FOSS approach can be successful and can produce software of high quality. On the other hand, the Mozilla Project supports many technologies, including popular tools such as the Firefox web browser and the Thunderbird e-mail client, among others. More details on these projects can be found in [5] and on the Mozilla website [30].

B. FOSS and the Power Systems Community

For the purpose of this paper, we have divided the power systems community in two areas: the Education and Research & Development area, which refers to research laboratories and universities; and the Power Business, which refers to power system analysis proprietary software developers and power utilities. This Section provides some outlines on how FOSS can be adopted by the power system community.

1) *FOSS in Education and R&D*: In the area of Education the FOSS approach has proved to be successful for teaching power system analysis. References [31] and [32] show that teaching using a FOSS can be effective and is well received by students. Unfortunately, to the knowledge of the authors, so far there has not been any specific application of FOSS Educational Learning Management Systems [33] for power system analysis.

An example of the use of FOSS for research is given in [34], which reflects PSAT users’ experience in this field. Moreover, FOSS is not a stranger to the R&D area at National Laboratories for power system analysis, an interesting example of the adoption of the OSS approach at PNNL is the GRIDLAB-D [35] simulation software part of the GridwiseTM Project

[36], this software incorporates novel modeling techniques and high-performance algorithms for power system analysis.

The Oak Ridge National Lab has developed a successful application that enables simulation of hybrid systems [37]. This is an example on how interdisciplinary research (in this case, computer networks and power systems) takes full advantage of the FOSS approach. As this type of interdisciplinary studies grow, flexibility will be a key feature. We believe that the FOSS approach can provide the required flexibility.

2) *FOSS and the Power Business*: Currently the main developers of power system analysis software have adopted the proprietary development model [38]–[44]. In the power system analysis community the *scripting* and *user defined modeling* features in proprietary software have often been misinterpreted as a sort of *open* feature of these type of software. Proprietary software companies are the only ones that can make changes or enhancements to the core software, they will charge for the end product as well as for the support, and the infringement of the software license is punishable by law.

The FOSS development method offers an alternative where small or big companies can charge for the support services of maintaining, modifying and improving software. This opens the service market for small software maintainers, and at the same time can trigger competition among these companies and the proprietary software counterparts. FOSS can be also a medium of publicity for a company. The example of OpenDSS, which is an EPRI product, is promising (see following subsection III-H). Companies developing power system analysis software interested in investigating the different business and economic models for FOSS are referred to Part IV of [5].

A frequent question raised by professionals in the power business is: Apart from research and development, to what extent can the power business take advantage of FOSS? In our view, the possibilities are countless and could cover the entire power business as we know it. Most professionals are likely already taking benefits from FOSS without realizing it. In a recent publication [45], KEMA analyzes in which way the OSS approach can be used for energy solutions at different levels of the Service-Oriented Architecture. The amount of opportunities is overwhelming and exciting. Interesting description on applying OSS to numerous fields of enterprises can be found at [10] and [11].

C. Software Licensing

Software licensing is a complex issue and software developers should study in detail what type of license is more suitable for their goals and vision. Licensing issues are of importance for developers, but generally not for the majority of users who will most likely never need to change the software. Our aim here is to provide a brief description of the categories of licenses and give references for further reading. For a detailed analysis of OS and FS licenses readers are referred to [9], additionally we point readers to [46] for an interest debate on software copyright.

1) *Free Licenses*: also called “reciprocal” licenses, contain a provisions that require that on re-licensing the code most remain open source. If the primary distributor obtains the

source code it should pass it down to others. In simple terms, if one chooses to distribute an application he has modified, one must include the source code for the derived work. Among many Free Licenses [3] the GPL is the most known.

2) *Open Licenses*: in contrast with the Free Licenses, Open Licenses or “non-reciprocal” licenses do not contain provisions on re-licensing, therefore they allow derivative works from open source code to be converted into closed software. Therefore, they are non reciprocal because when the primary recipient obtains the source code he may or may not pass it on. The OSI maintains a list of Open Licenses [4], the BSD and MIT licenses are among the most popular.

3) *Non-Open Source Licenses*: classic proprietary licenses need little explanation. The license does not make a distinction between the source code and the binary because the source code is not made available. The license does not allow to distribute the software [9].

4) *Other Licenses*: There are many other licenses like the Qt License [47], the Artistic License [48], and the Creative Commons Licenses [49]. These licenses have very unique features that reflect the terms that the originators decided to impose on users and modifiers of their work. Unlike the Free and Open licenses, these are mostly only applied to the programs that they where developed for, and they are adapted for the code originally licensed under them and their derivative works [9].

III. FOSS FOR POWER SYSTEM ANALYSIS

This section describes the main free and open source projects for power system analysis. Subsection III-I presents a list of other minor projects, most of which are not maintained any longer. The descriptions below do not pretend to be exhaustive and we invite the reader to visit the web page of each project.

Some of the projects that are described in this section have been widely used by a great number of undergraduate and Ph.D. students all around the world and have greatly helped in increasing the knowledge on power systems.

Unfortunately, most of these packages are maintained by one or few researchers. Synergies between different packages are almost absent, which is a pity, but maybe the Task Force on Open Source Software can change this situation. In addition, contributions to the development of these packages by users are rare. This fact is uncommon in open source software, but can be partially explained by the fact that the users are students of electrical engineering but are not experienced programmers.

Table I summarizes some relevant features of the FOSS that are described in this section. The features are: programming language, main purpose, interface type, and license type.

A. UWPFLOW and TEFTS (1996)

UWPFLOW is a research tool that has been designed to calculate local bifurcations related to system limits or singularities in the system Jacobian [13]. UWPFLOW also generates a series of output files that allow further analyses, such as tangent vectors, left and right eigenvectors at a singular bifurcation point, Jacobians, power flow solutions

TABLE I
FEATURES OF EXISTING FOSS FOR POWER SYSTEM ANALYSIS

Name	Language	Purpose	Interface	License
UWPFLOW	C	Continuation power flow	UNIX-like command line	Open license
TEFTS	C	Transient stability	UNIX-like command line	Open license
MatPower	MATLAB	Optimal power flow	MATLAB functions	Open license
PSAT	MATLAB, GNU Octave, Perl	Static and dynamic stability analysis	MATLAB GUIs and Simulink	GNU GPL
VST	MATLAB	Voltage stability	MATLAB GUIs	Open license
InterPSS	Java	Power system analysis for industry applications	Java GUI	GNU GPL
AMES	Java	Whole sale power markets	Java command line	GNU GPL
DCOPFJ	Java	Dc optimal power flow	Java command line	GNU GPL
Pylon	Python	Dc and ac power flow and dc optimal power flow	Graphviz and Chaco	GNU GPL
OpenDSS	Delphi	Distribution system planning	Delphi GUIs	BSD license

at different loading levels, voltage stability indices, etc. The latest version of UWPFLOW dates April 2006. TEFTS is a simple transient stability program to research Transient Energy Function (TEF) and voltage stability issues in dynamic models of AC-HVDC systems [14]. Both programs are written in C, and two versions for DOS (16 bit) and UNIX are available, including examples and brief tutorials. The tools are maintained by Prof. C. A. Cañizares, University of Waterloo, Ontario, Canada.

B. MatPower (1997)

MatPower is a package of MATLAB m-files for solving power flow and optimal power flow problems [15]. It is intended as a simulation tool for researchers and educators that is easy to use and modify. MatPower is designed to give the best performance possible while keeping the code simple to understand and modify. Although MatPower is free and open source, it runs on MATLAB which requires a license. The application is developed by the E&CE group of the Cornell University, Ithaca, New York.

C. VST (2002)

Voltage Stability Toolbox (VST) has been developed to investigate stability and bifurcation issues in power systems [16]. VST integrates the symbolic and numeric computations with a graphical menu-driven interface based on MATLAB and its Extended Symbolic Toolbox. VST implements symbolic computations to build exact load flow equations and Jacobian matrices including 2nd order derivatives, required for bifurcation analysis. Numerical calculations of solutions to power system equations are performed and controlled via a graphical user interface (GUI).

D. PSAT (2002)

The Power System Analysis Toolbox (PSAT) is a MATLAB and GNU/Octave-based toolbox for electric power system analysis [17]. Main features of PSAT are: power flow; continuation power flow; optimal power flow; small signal stability analysis; time domain simulation; Phasor Measurement Unit (PMU) placement; user defined models; FACTS Models; wind turbine models; conversion of data files from several formats; export results to EPS, plain text, MS Excel and \LaTeX files;

and interfaces to GAMS and UWPFLOW Programs. PSAT provides also a complete GUI and an one-line diagram editor (see Fig. 1). The new version of PSAT, which has been released as version 3, is based on Python and deeply exploits modern concepts of this powerful object-oriented script-based language. PSAT is maintained by Prof. F. Milano, University of Castilla-La Mancha, Spain.

E. InterPSS (2006)

The Internet Technology-based Open-source Power System Simulation System (InterPSS) is a Java-based, open-source development project aimed to develop a simple to use, yet powerful Internet technology based software system for design, analysis, and simulation of power systems [18]. Its open and loosely coupled system architecture allows third-party components to be easily plugged into the system to augment its functionality, and allows its components to be integrated into other systems to provide certain power system simulation functionality or services (see Fig. 3). The project is currently under development by a team of developers from the United States, Canada and China.

F. AMES and DCOPFJ (2007)

The AMES and DCOPFJ are FOSS packages developed entirely in Java by an interdisciplinary team of researchers at Iowa State University [19], [20]. AMES is an extensible and modular agent-based framework for studying the dynamic

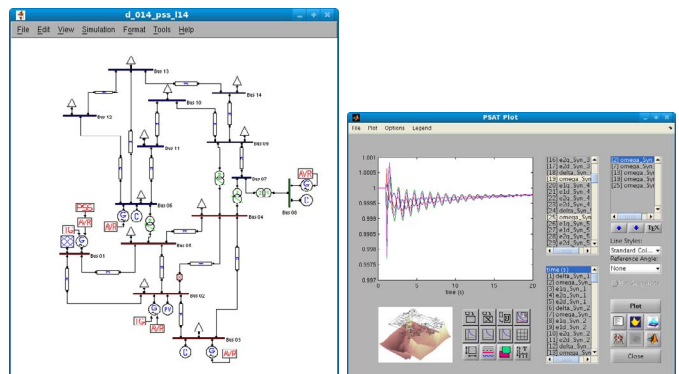


Fig. 1. PSAT GUIs.

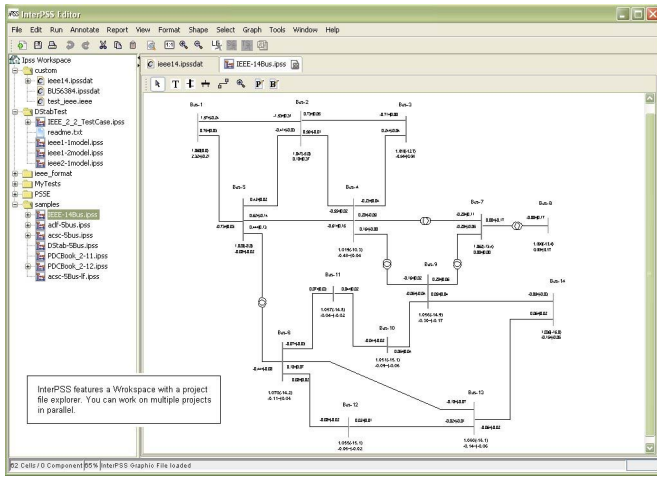


Fig. 2. InterPSS main GUI.

efficiency and reliability of wholesale power markets, while DCOPFJ is a solver for small to medium-sized DC optimal power flow problems having a strictly convex quadratic programming (SCQP) formulation. These programs also provide a graphical user interface and output reports through table and chart displays.

G. Pylon (2007)

Pylon is a general purpose, open standards driven, cross-platform tool for simulation of electric power systems based on Python [21]. Its current features include: dc power flow, ac power flow and dc optimal power flow. Pylon is based on Traits (<http://code.enthought.com/traits>) that allow easily defining and extending power system data models (see Fig. 3). Calculations are based on CVX-OPT (<http://abel.ee.ucla.edu/cvxopt>) that links Python to efficient libraries such as LAPACK and UMFPACK to enhance computational performance. Pylon can be easily extended through scripts or embedded in other applications. Pylon is maintained by R. W. Lincoln, University of Strathclyde, UK.

H. OpenDSS (2008)

The OpenDSS is a simulator specifically designed to represent electric power distribution circuits [22]. OpenDSS is designed to support most types of power distribution planning analysis associated with the interconnection of distributed generation (DG) to utility systems. It represents unbalanced conditions, stochastic processes, and other aspects of electrical power distribution systems including detailed equipment models. The user can extend the main programs and add user defined models. OpenDSS is maintained by a team of engineers working with EPRI.

I. Other FOSS Projects

In this subsection we list some free (but not always open source) projects that are available on the Internet or that can be obtained from the authors. Some of these packages are not

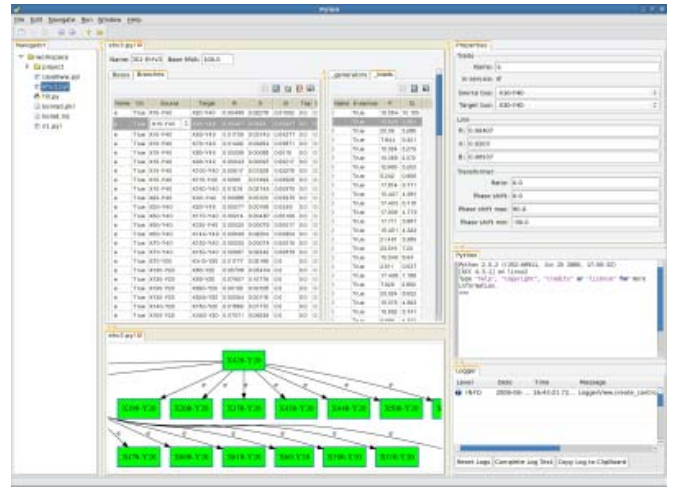


Fig. 3. Pylon main GUI.

maintained any longer. A list of other public domain projects can be found at [50].

- 1) *ObjectStab* is a free Modelica Library for Power Systems [51]. Contrary to special-purpose tools for the same type of simulations like PSS/E, Eurostag and SIMPOW, the ObjectStab library has an open architecture so that all models can be customized to fit the individual users' needs. The library is intended for research and educational use.
- 2) *Power Education Toolbox* (PET) contains a program for power flow, state estimation (WLS and LAV) and network observability analysis functions. The zip file containing the program, help files, and a small 14-bus test system. PET is available at <http://ee.tamu.edu/~abur/pet.html>
- 3) *FlowDemo.net* is an interactive, Internet and Java based application to visualize power flow in an electric transmission system. The demonstration program allows the user to open switches, change loads, and change generation in 6 predefined topologies. FlowDemo.net is available at <http://flowdemo.net/>
- 4) *Power Analysis Toolbox* (PAT) [52] and the *Education Simulation Tool* [53] are two similar projects based on MATLAB/Simulink, that allow easily modeling synchronous machines controls such as AVRs, turbine governors and PSSs.

IV. IEEE TASK FORCE ON OSS

The Task Force in Open Source Software for Power Systems explores the potential for FOSS in the Power Engineering Society (PES). The Task Force began its activities in Fall 2008. The mission of the Task Force is twofold: (i) to diffuse the philosophy of FOSS in the power systems community; and (ii) promote FOSS for the benefit of the PES ranging from pedagogical purposes to commercial-grade applications.

The whole idea of FOSS is to encourage others to join and contribute. We believe that the most important objective for this Task Force is to get people interested and involved in FOSS.

For this purpose, the TF has created its own web-page where the maintainers collect an as-complete-as possible list of free and/or open source software packages for power system analysis [54]. To better diffuse the FOSS philosophy, the Task Force is also preparing a report that describes the state of the art of FOSS for power systems analysis. This paper is a preliminary draft for the report. The report will also summarize the contributions of the panel session on FOSS held at the IEEE PES General Meetings, Calgary 2009.

V. FUTURE OF FOSS FOR POWER SYSTEMS

The future of FOSS for power system analysis is promising although not completely clear. One of the most difficult tasks is to convince researchers and practitioners to share their knowledge and code. This practice is common in several scientific fields but unfortunately rare in the power system community. In the following, we list some features of FOSS that can be the key of its success in power system applications.

1) *Education and research takes advantage from FOSS:* In our experience, the students generally accept with enthusiasm the ideas from FOSS and tend to diffuse them once they begin their professional career. However, to achieve this goal, professors should adopt FOSS ideas first.

2) *FOSS is a business:* FOSS should not be limited to universities or idealists, it can also be a business. Several open source projects have become important companies. We believe that there is room for this kind of business also in power systems. Projects such InterPSS and OpenDSS are pioneers in this approach. InterPSS provides consulting services under request, while OpenDSS is used as a publicity platform for other EPRI products. The product is the know-how and the experience of the people that maintain the software not the software itself.

3) *FOSS reduces costs and improves reliability:* A final remark on the advantages of FOSS that could incentive companies to provide their products under a GPL license or similar is the fact that FOSS projects typically have a large number of users. This basically provides an invaluable testing platform. One of the most important issues of software applications for power system analysis is that due to the reduced number of users the testing of these applications is very costly.

VI. CONCLUSIONS

The main concepts of FOSS that are relevant for power system analysis, research and education are introduced and discussed, as well as the main FOSS projects for power system analysis. The paper also provides some outlines about the future of FOSS for power systems and the activities of the IEEE Task force on Open Source Software.

It is our hope that promoting FOSS ideas in the power system community can help diffusing knowledge and improving the development of power systems analysis. Any comments, suggestion and contributions of the IEEE Task Force for Power Systems are very welcome.

REFERENCES

- [1] R. M. Stallman, *Free Software, Free Society: Selected Essays of Richard M. Stallman*. Boston: Free Software Foundation, 2002.
- [2] "Free Software Foundation (FSF)," available at <http://www.fsf.org/>.
- [3] R. M. Stallman, *GNU General Public License*. Free Software Foundation, 2007, available at <http://www.gnu.org/copyleft/gpl.html>.
- [4] "Open Source Initiative (OSI)," available online <http://www.opensource.org/>.
- [5] J. Feller, B. Fitzgerald, S. A. Hissam, and K. R. Lakhani, Eds., *Perspectives on Free and Open Source Software*. Cambridge, MA: MIT Press, 2005.
- [6] C. DiBona, S. Ockman, and M. Stone, Eds., *Open Sources: Voices from the Open Source Revolution*. Sebastopol, CA: O'Reilly & Associates, 1999.
- [7] C. DiBona, D. Cooper, and M. Stone, Eds., *Open Sources 2.0: The Continuing Evolution*. Sebastopol, CA: O'Reilly Media, 2006.
- [8] R. W. Hahn, Ed., *Government Policy toward Open Source Software*. Washington, D.C.: AEI-Brookings Joint Center for Regulatory Studies, 2002.
- [9] A. M. S. Laurent, *Understanding Open Source & Free Software Licensing*. Sebastopol, CA: O'Reilly Media, 2004.
- [10] P. Kavanagh, *Open Source Software: Implementation and Management*. Oxford, UK: Elsevier Digital Press, 2004.
- [11] D. Woods and G. Guliani, *Open Source for the Enterprise: Managing Risks, Reaping Rewards*. Sebastopol, CA: O'Reilly Media, 2005.
- [12] S. Chopra and S. D. Dexter, *Decoding Liberation: The Promise of Free and Open Source Software*. New York: Routledge Taylor & Francis Group, 2008.
- [13] C. A. Cañizares and F. L. Alvarado, "UWPFLOW Program," 2000, university of Waterloo, available at <http://www.power.uwaterloo.ca>.
- [14] —, "TEFTS Program," 2000, university of Waterloo, available at <http://www.power.uwaterloo.ca>.
- [15] R. D. Zimmerman and D. Gan, *Matpower, Documentation for version 2*, Power System Engineering Research Center, Cornell University, 1997, available at <http://www.pserc.cornell.edu/matpower/matpower.html>.
- [16] C. Nwankpa, *Voltage Stability Toolbox, version 2*, Center for Electric Power Engineering, Drexel University, 2002, available at <http://power.ece.drexel.edu/research/VST/vst.htm>.
- [17] F. Milano, "An Open Source Power System Analysis Toolbox," *IEEE Transactions on Power Systems*, vol. 20, no. 3, pp. 1199–1206, Aug. 2005.
- [18] M. Zhou, "InterPSS," available at <http://www.interpss.org>.
- [19] "AMES Market Package," available at <http://www.econ.iastate.edu/tesfatsi/AMESMarketHome.htm>.
- [20] "DCOPFJ Package," available at <http://www.econ.iastate.edu/tesfatsi/DCOPFJHome.htm>.
- [21] "Pylon, Power system and energy market analysis with Python," <http://pylon.eee.strath.ac.uk/pylon>.
- [22] "OpenDSS," available at <http://sourceforge.net/projects/electricdss/>.
- [23] "IEEE Task Force on Open Source Software," available at http://ewh.ieee.org/cmte/psace/CAMS_taskforce/index.htm.
- [24] U.S. Congress. Senate, "Digital Millennium Copyright Act," available at <http://thomas.loc.gov/>.
- [25] L. Lessing, *The Future of Ideas: The Fate of the Commons in a Connected World*. Vintage, 2002.
- [26] K. Coar, "Open Source Definition," 2007, available at <http://www.opensource.org/docs/osd>.
- [27] "FSF Free Software Licensing and Compliance Lab," available at <http://www.fsf.org/licensing/>.
- [28] E. S. Raymond, *The Cathedral and the Bazaar*. Thyrsus Enterprises, 2000, available at: <http://www.tuxedo.org/esr/>.
- [29] "Apache Software Foundation," available at <http://www.apache.org/>.
- [30] The Mozilla Foundation, "mozilla.org," available at <http://www.mozilla.org/>.
- [31] L. Vanfretti and F. Milano, "Application of the PSAT, an Open Source Software, for Educational and Research Purposes," in *IEEE PES General Meeting*, Tampa, USA, June 2007.
- [32] F. Milano, L. Vanfretti, and J. C. Morataya, "An Open Source Power System Virtual Laboratory: The PSAT Case and Experience," *IEEE Transactions on Education*, vol. 51, no. 1, pp. 17–23, Feb. 2008.
- [33] S. E. Lakhani and K. Jhunjhunwala, "Open Source Software in Education," *EDUCAUSE Quarterly*, vol. 31, no. 2, April-June 2008, available online: <http://net.educause.edu/ir/library/pdf/EQM0824.pdf>.

- [34] L. Vanfretti and F. Milano, "The Experience of PSAT (Power System Analysis Toolbox) as a Free and Open Source Software for Power System Education and Research," *International Journal of Electrical Engineering Education*, vol. 43, no. 3, July 2009, Accepted for future publication. Available at <http://www.power.uwaterloo.ca/~fmilano/publications.htm>.
- [35] D. Chassin, K. Schneider, and C. Gerkensmeyer, "Gridlab-d: An open-source power systems modeling and simulation environment," *Transmission and Distribution Conference and Exposition, 2008. T&D. IEEE/PES*, pp. 1–5, April 2008.
- [36] PNNL, "GridwiseTM at PNNL," available at <http://gridwise.pnl.gov/>.
- [37] J. Nutaro, P. T. Kuruganti, L. Miller, S. Mullen, and M. Shankar, "Integrated hybrid-simulation of electric power and communications systems," *IEEE Power Engineering Society General Meeting 2007*, pp. 1–8, June 2007.
- [38] "Siemens PTI (Power Technologies International)," available at <http://www.pti-us.com>.
- [39] "PowerWorld Corporation," available at <http://www.powerworld.com>.
- [40] G. Energy, *GE-PSLF Load Flow Data Export/Import File for PSLF Version 15.1*, General Electric International, Inc., 1 River Road, Schenectady, NY 12345, USA, June 2005.
- [41] "CYME International Inc." available at <http://www.cyme.com>.
- [42] "NEPLAN," available at <http://www.neplan.ch>.
- [43] "ABB Simpov," available at <http://www.stri.se>.
- [44] "Eurostag," available at <http://www.eurostag.be>.
- [45] S. Fustar, "The impact of open source software on the next generation of energy systems," *IEEE Power Engineering Society General Meeting 2007*, pp. 1–1, June 2007.
- [46] A. G. González, "The software patent debate," *Journal of Intellectual Property Law & Practice*, vol. 1, no. 3, pp. 196–2006, Jan 2006.
- [47] Nokia Qt Software (Formerly Trolltech ASA), "Qt licensing," available at <http://trolltech.com/products/appdev/licensing>.
- [48] The Perl Foundation, "Artistic License 2.0," August 1997, available at http://www.perl.org/artistic_license_2.0.
- [49] Creative Commons Corporation, "Creative Commons Licenses," available at <http://creativecommons.org/licenses/>.
- [50] "Digital educational resources for power systems," available at http://www.ece.mtu.edu/faculty/ljbohman/peec/Dig_Rsor.htm.
- [51] M. Larsson, "ObjectStab – An Educational Tool for Power System Stability Studies," *IEEE Transactions on Power Systems*, vol. 19, no. 1, pp. 56–63, Feb. 2004.
- [52] K. Schoder, A. Hasanović, A. Feliachi, and A. Hasanović, "PAT: A Power Analysis Toolbox for MATLAB/Simulink," *IEEE Transactions on Power Systems*, vol. 18, no. 1, pp. 42–47, Feb. 2003.
- [53] C. D. Vournas, E. G. Potamianakis, C. Moors, and T. Van Cutsem, "An Educational Simulation Tool for Power System Control and Stability," *IEEE Transactions on Power Systems*, vol. 19, no. 1, pp. 48–55, Feb. 2004.
- [54] "IEEE Task Force on Open Source Software for Power Systems," available at http://ewh.ieee.org/cmte/psace/CAMS_taskforce/index.htm.



Federico Milano (M'03) received from the University of Genoa, Italy, the Electrical Engineering degree and the Ph.D. degree in 1999 and 2003, respectively. From 2001 to 2002 he worked at the University of Waterloo, Canada, as a Visiting Scholar. He is currently an assistant Professor at the University of Castilla-La Mancha, Ciudad Real, Spain. His research interests include voltage stability, electricity markets and computer-based power system modelling and analysis.



Luigi Vanfretti (S'03) received the Electrical Engineering degree from *Universidad de San Carlos de Guatemala* in 2005. He is a Ph.D. student in the Electrical, Computer, and Systems Eng. Dept. at Rensselaer Polytechnic Institute, from where he obtained his MS in 2007. He was a visiting researcher at the Dept. of Electronics and Electrical Eng. of the University of Glasgow, Scotland, in 2005. His research interests are modeling, dynamics, and stability of power systems; applications of PMU data, and open source software for power system analysis.