Pedagogical Interventions through Software tools in Postgraduate Engineering Programme

R. P. Tapaskar[1], P. P. Revankar[2], M. B. Gorwar[3], R. S. Hosmath[4]

School of Mechanical Engineering Sciences,
K. L. E Technological University, Hubballi
[1]rptapaskar@bvb.edu
[2]pp_revankar@bvb.edu

Abstract: Educational institutes have been practicing the proprietary software extensively with a profound impression that only the proprietary software can accomplish the educational requirements leading anticipated outcomes. In fact, open source software (OSS) tools can provide the indispensable flexibility to combine languages, scripts, learning objects and lesson plans, effectively, without the cost and rigidity of proprietary software and other added benefits. The present pedagogical intervention highlights the idea computational tools and software in postgraduate engineering curriculum in Energy Systems Engineering. The proposed work on the available options which can potentially employ Software tool with benefits to educational institutes and student's community. The outcomes of implementing these pedagogical strategies have shown the accelerated learning skills of the students in the engineering discipline.

Keywords: Open source tools and software, Engineering education.

I Introduction

Academia is one of the potential dais for educating engineering pupils and developing finished commodities which are future open source developers. Educating the pupils of undergraduate engineering to work on open source software tools and software leads to future engineers who will be acquainted with the developer tools instead of finished proprietary software intended for the use of laymen's and commercial consumers. Software is a sequence of instructions that directs the computer to perform specific tasks or operations. The role of software tools and simulators are well evident in engineering education and research. They also play a vital role in the development of wide variety of complex engineering system. The same holds good at the industrial perspective where the OSS product which is developed in-house is not provided to any community outside the organization, but it is handled as an open source project within the company serving the custom requirements of the company [1].

Introduction of open source software tools in the PG Energy Systems engineering programme has brought up an upright change in the learning skills of students inclining them to think as the developers and engineers during use of the tools rather than be a mere end user of the proprietary software tools. Scilab tool in the renewable energy lab was used to solve complex the mathematical computations rather than proprietary software like MatLab. Scilab not only served the purpose of complex engineering computation and programming but at the same time students appreciated the ease of use and features that Scilab offered during the learning phase of the software even being a freeware. Another important open source tool for the word processing and presentation used was Libre office package which successfully replaced traditional Microsoft's office package without any exertion in adoption in the daily usage. Libre office could replace the proprietary alternative and also made the learning a fun experience to students where they explore the utility of the software rather than user centric ready-solutions available proprietary software tools which would hinder their exploration-intellectual instinct.

Computer software is a series of instructions commands that dictate a computer to perform specific tasks or operations it consists of computer programs, libraries and related non-executable data. Software can be broadly divided into following types based on the nature of intended usage.

Proprietary software: Software licensed under the exclusive legal rights of the copyright holder and user should pay the registration or licensing fee but imposed restrictions to change the source code.
Freeware: Software without payment but usually cannot be modified, re-distributed or reverse engineered without authors permission.

Shareware: Software provided with free of cost but users are encouraged to share the copies of program.
Adware: Software automatically renders the advertisements in order to generate revenue for its author when installed on the client system.
Demoware: Demonstration version of software with two types - crippled and trial ware.
Cripple ware: Most of the features are enabled but disabling the utility features such as printing or storing of the file.
Trialware: Software with trial period and automatically stops functioning, unless the user pays the license fee to turn it registered version.
Donationware: Fully operational software with request for optional donation to programmer or third-party beneficiary.
Nagware: Software persistently reminds or nags the user to register it by paying a fee by popping up a message.

Freemium: Software works by offering a product or service free of charge, while charging a premium for advanced features, functionality or related products and services.
Open source software: It is computer software with its source code made available and licensed with an open-source license in which the copyright holder provides the rights to study, change and distribute the software for free to anyone and for any purpose. Open-source software is very often developed in a public, collaborative manner. Open-source software is the most prominent example of open-source development and often compared to (technically defined) user-generated content or (legally defined) open-content movements.
Public domain software: it is the software tool which is kept in the public domain. There is no any one's ownership such as copyright, trademark or patent on such type of software. These can be modified, distributed or sold even without any attribution by anyone.

II Literature review:

Numerous research has been encountered in the use of open source tools in academia with exhaustive analysis on the benefits for institutes and students, the section summaries the research carried in relevance with open source-academia programs.

A review of electronics engineering design free software tools presented by Vijay Nehra et al. [3] compared existing open source tools in reference to the factors like simulations, schematic capture, PCB and HDL designs. Researchers concluded efforts should focus on the existing projects rather than new one.

N. Pankaja et al. [4] presented a framework for integrating free open source tools and resource as teaching resources for an electronic engineering undergraduate program for improving the learning outcomes of the students and the author give a conclusion that the free/open source software are the promising utilities for both to widen up their skills and research.

Scott Teel et al. [5] the best suited alternatives for the proprietary software in undergraduate electronic program are discussed with the analysis of impact of open source software if the undergraduate pupil and mentors adopt them into their study.

Walt Scacchi [6] made the comparison between market monopolization and industry standards of open source software are discussed and also the author concluded that there should be a collaborative contribution by both academia and industries in the current pace so as to achieve a whole new scenario.

In reference [7] the pro and cons of both proprietary and open source software are mentioned without a clear winner. The different lead-lag situations are discussed for the proprietary and open source tools concluding that they are different with each other based on different philosophies, methodologies and business models.

More over as time passes developer community expands and the developed tools are free and support casual use, more members of the development team will be able to access and contribute to artifacts in all phases of development [2].

To summarize the literature section, we can conclude that various researches have been reported by employing the open source tools and software in academia as well as industrial sectors. Some have adopted the whole package for the computations viz. operating system and also the application software, where as some have adopted partially by installing the open source computational tools on the proprietary operating system. One of the major challenge faced during the initial use of the open source tool is the learning time incurred while transacting from proprietary to open source tools.

III. Strategies for Post Graduate Programme

The computers are today the integral part of the engineering profession and provide realistic and fast solutions to real time problems. The Industry strongly recommends the competencies in computer based skills that enable engineering graduates to design systems more effectively. The graduate attribute PO5 emphasizes on computer competencies in graduates so as to prepare them for industry oriented tasks that have been more and more computer linked. The scientific world today has evolved computer programme for practically all physical systems that have a defined governing law and hence their behavior can be studied through these tools. The process of imitation of reality- termed as Simulation is a powerful tool irrespective of the discipline of engineering be it flow visualization, circuit analysis or chemical kinetics. The simulation has been recognized as an effective stage of engineering design by the corporate world that envisages innovation and optimization of resources. The concept blends the various governing laws of the physical process into a packaged solution to real time engineering problems. The introduction of computer based system design integrates technical knowledge with the optimization techniques to provide a realistic solution that would lead to a competitive product. If the concept of technical design and optimization process are studied in isolation the student will not be capable to appreciate either of the two. Therefore, computer based curricula definitely gives an opportunity to expand the horizon of the student thought process. The curricula delivered for the PG programme in Energy systems engineering takes account of this aspect and the course delivery is made such that each course has a component of computer usage. The rubrics defined for the course took cognizance of motivating students in the usage of computational tools for the theoretical concepts they learned. The following lists of software were available to students for execution of course connected assignment or the Project work. The courseware was grouped as operating systems, mathematical computational tools and simulation tools.

a) Operating System

Linux is Unix-like computer operating system (OS) assembled under model of free and open-source software development and distribution. The defining component of Linux is the Linux kernel an operating system kernel first released on 5 October 1991 by Linus Torvalds. The development of Linux is one of the most prominent examples of free and open-source software collaboration. The underlying source code may be used, modified and distributed, commercially or non-commercially, by anyone under the terms of its respective licenses. Linux is analogous to popular proprietary operating system like windows and Macintosh.

Office Software

LibreOffice is most actively developed free and open source office suite, a project of The Document Foundation. It was forked from OpenOffice.org in 2010, which was an open-sourced version of the earlier Star Office. The LibreOffice suite comprises programs for word processing, the creation and editing of spreadsheets, slideshows, diagrams and drawings, working with databases, and composing mathematical formulae. It is available in 110 languages. LibreOffice is available for a variety of computing platforms, including Microsoft Windows, OS X (10.8 or newer), and Linux. It is the default office suite of most popular Linux distributions including packages viz. Writer, Calc, Impress, Draw, Math and Base which can be alternative to proprietary Microsoft Office Package.

b) Mathematical computation tools

Scilab is an open source, cross-platform numerical computational package and a high-level, numerically oriented programming language. It can be used for signal processing, statistical and numerical analysis, image enhancement, fluid dynamics simulations, numerical optimization, and modelling, simulation of explicit and implicit dynamical systems and (if the corresponding toolbox is installed) symbolic manipulations. Scilab is one of the two major open-source alternatives to MATLAB, the other one being GNU Octave. Scilab is similar enough to MATLAB that some book authors (who use it) argue that it is easy to transfer skills between the two systems. Scilab however puts less emphasis on (bidirectional) syntactic compatibility with MATLAB than Octave does.

c) Simulation tools

Computer Aided Design

Google Sketch Up is an open source AutoCAD alternative. SketchUp has been lauded as a very versatile and easy-to-use alternative for design students and professionals.

HDLVERILOG Simulator

Verilator is a free and open source software tool which converts Verilog (a hardware description language) to a cycle-accurate behavioural model in C++ or SystemC. It is restricted to modelling the synthesizable subset of Verilog and the generated models are cycle-accurate, 2-state, with synthesis (zero delay) semantics. As a consequence, the models typically offer higher performance than the more widely used event driven simulators, which can process the entire Verilog language and model behaviour within the clock cycle. Verilator is now used within academic research, open source projects and for commercial semiconductor development. It is part of the growing body of free EDA software.

PCB Design

KiCad is a free software suite for electronic design automation (EDA). It facilitates the design of schematics for electronic circuits and their conversion to PCB designs. KiCad was originally developed by Jean-Pierre Charras, and features an integrated environment for schematic capture and PCB layout design. Tools exist within the package to create a bill of materials, artwork, Gerber files, and 3D views of the PCB and its components.

8051 Simulator

MCU 8051 IDE is a new modern graphical integrated development environment for microcontrollers based on 8051. For those who believe 8051 is a great piece of technology this IDE is a new way how to see and feel these still famous microcontrollers. **MCU 8051 IDE is open-source** software primarily for Microsoft Windows and GNU/Linux licensed under the terms of GNU GPLv3 license. One can use this IDE for education, business, hobby, or something else, for Linux this software stays **completely free of charge**, and for Windows there is a small fee for a copy in order make it possible to maintain the software and provide technical support for it.

VLSI Design

Magic is a venerable VLSI layout tool, written in the 1980's at Berkeley by John Ousterhout, now famous primarily for writing the scripting interpreter language Tcl. Due largely in part to its liberal Berkeley open-source license, magic has remained popular with universities and small companies. The open-source license has allowed VLSI engineers with a bent toward programming to implement clever ideas and help magic stay abreast of fabrication technology. However, it is the well-thought-out core algorithms which lend to magic the greatest part of its popularity. Magic is widely cited as being the easiest tool to use for circuit layout, even for people who ultimately rely on commercial tools for their product design flow.

Circuit Simulator

Ngspice is a mixed-level/mixed-signal circuit simulator. Its code is based on three open source software packages: Spice3f5, Cider1b1 and Xspice. It is the open source successor of these venerable packages. Many, many modifications, bug fixes and improvements have been added to the code, yielding a stable and reliable simulator. Therefore, besides being used as a standalone simulator, Ngspice has been incorporated into many projects, see our simulation environments page.

IV Advantages and Disadvantages of OSS

The use of OSS was encouraged amongst the students to ensure that students got access to OSS on their personal computing devices without investment of any money. There were certain issues related to the difficulty in use of OSS that had poor user interfaces. The important issues related to use of the OSS has been discussed in the following section of the publication.

a) Advantages of OSS

Learn concepts, instead of train for products

Objective of Academic institutes concentrate more on teaching and learning of concepts instead of training of products, students have to be trained to learn basic concepts behind the use of products rather than just use the ready product. Also, open source software facilitates customizations which teachers can modify to suit the context.

Absence of license fee

Academia will pay huge amount yearly as the license pay for the proprietary software. But open sourcetools are free of cost and license fee is absent which provides added advantage on the academic institution in reducing the recurring cost of license renewal for proprietary software.

Continues improvement

Developers for open source software are constantly trying to improvise the software performance with time to be competent enough with the propriety software which is another advantage leading to be updated with time.

Prevention of Illegal Copying

Students can install the Open Source Software tool in their own computers without any restriction and thus illegal copying of software is not necessary for their academic work outside institution premises.

Disadvantages of Open Source Software

There are some of the disadvantages associated with the open source software, some of which are enlisted.

Lower Security

Open source products are developed by the open market free developers. These are not secure enough as compared to the proprietary counterparts, resulting in the usage of these open source tools in the data sensitive critical applications such as financial and medical records.

No Free Support

The open source tools are more developer oriented rather than consumer orientated, if the user faces any technical issue with the software than user has to spend time in resolving the issue because of the absence of technical support team as that of proprietary software.

V Computation Fluid Dynamics course

The course was taught at the second semester of the masters Programme in Energy Systems Engineering with a component of theory and lab section. The course provided the necessary background on the physics of fluid flow through conceptualization of Navier Stokes Equation and its computational aspects strengthened the concepts learnt in the theory course. This course was further useful for the execution of Project work prescribed during the third and fourth semesters of the master's programme. The results of the course in terms of the combined assessment of In Semester Assessment and End Semester Assessment are presented in figure 1 to demonstrate the flair students have in computing tools as compared to conceptualization of the course through theoretical discussions. The observations clearly indicated that average distribution of grades in the lab component was 27.99% higher as compared to the performance in theory component that respectively showed as average Grade point average of 8.87 and 6.93 on a 10-pointer scale for a 15-student batch size.

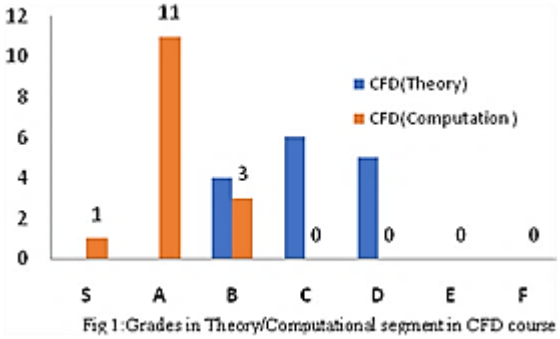

Fig 1:Grades in Theory/Computational segment in CFD course

The later observations of student performance in the programme revealed that computational skills imbibed in early part of the master's programme were of immense help during later stage of the Programme for execution of project work or to get placement opportunities. The presented case is an isolated example of implementing computational tools in the master's programme. The other courses also had their own way of introducing the computational aspects in the curriculum. In view of the observations made in the curriculum delivery it was strongly suggested to introduce a many computational software as possible so as to provide a real-time exposure of the course content along with assimilating the theoretical rudiments. The introduction of computer related tasks also motivated students to take-up very interesting real-time projects that included data-acquisition systems, simulations on solar thermal systems and development of low cost computing tools. The use of this strategy also improved the number of technical publications by the student at various competitions.

VI Conclusions

The observations drawn from the presented work strongly indicated that free and open source software are cheap, secure and customizable software that have wide application in making the world less dependent of the proprietary software and to innovate new products to add knowledge to the domain of scientific community.

The role of open source in electronics industry is broadening, paving a career path for enthusiast. There is an increasing demand for electronics engineers who are experienced in open source technologies, mainly due to a change in development towards a more social and collaborative development environment.

The computational tools are targeted towards attainment of PO5 of the Graduate attributes along with PO4 that relates to doing research problem synthesis

The computational tools have improved quality of research as presented in master's student thesis of Energy Systems Engineering Programme.

VII References

[1] Martin Host, Alma Orucevic-Alagic, A Systematic Review of Research on Opensource Software in Commercial Software Product Development.

[2] Jason E Robbins, Adopting Open Source Software Engineering (OSSE) Practices by adopting OSSE Tools.

[3] Vijay Nehra, Aruna Tyagi, Free Open Source Software in Electronics Engineering Education: A Survey. I. J Modern Education and Computer Science, 2014.

[4] N. Pankaja, Mukund Raj P K, Proprietary software versus Open Source Software for Education, American Journal of Engineering Research (AJER) e-ISSN: 2320-0847 p-ISSN: 2320-0936 Volume-02, Issue-07, pp-124-130.

[5] Scott Teel, Dino Schweitzer, and Steve Fulton, Teaching Undergraduate Software Engineering Using Open Source Development Tools, Issues in Informing Science and Information Technology Volume 9, 2012.

[6] Walt Scacchi, The Future of Research in Free/Open Source Software Development

[7] https://en.wikipedia.org/wiki/Verilator

[8] https://en.wikipedia.org/wiki/KiCad

[9] http://www.moravia-microsystems.com/mcu-8051-ide/

[10] http://opencircuitdesign.com/magic/

[11] http://ngspice.sourceforge.net/presentation.html

[12] https://en.wikipedia.org/wiki/LibreOffice

[13] https://en.wikipedia.org/wiki/Scilab

[14] https://en.wikipedia.org/wiki/Linux

[15] http://blog.cometdocs.com/10-great-free-alternatives-to-autocad