A Temporal Visual Distribution of Learning Activities in FLOSS Repositories

Patrick Mukala, Ph.D

Department of Computer Science
University of Pisa, Italy
{patrick.mukala}@di.unipi.it

Abstract. Process mining is a relatively new field that encompasses powerful data and process analytics techniques for understanding processes from event data. In addition to these main techniques, it provides means enabling a pictorial representation of the occurrence of events over time. By applying such visualizations to event data from Free/Libre Open Source Software (FLOSS) environments, we get a complete understanding of how certain activities take place within such environments over time. Particularly, given the increasing interests in learning paradigms present in FLOSS communities, we believe that a temporal visual representation of learning events can yield great benefits. In this paper, we make use of the dotted chart in process mining to model and present a representation of learning behaviours over time for FLOSS participants.

Keywords: Dotted Chart, FLOSS Data, Educational Data Mining, Learning in FLOSS, Learning Visualization, Learning Analytics

1 Introduction

A considerable number of studies have provided critical insights pertaining to the existing opportunities and knowledge exchange in Free/Libre Open Source Software (FLOSS) environments [2–5,8,13,24,26].Some of these studies establish FLOSS communities as environments where successful collaborative and participatory learning between participants occurs [3,5,13]. These insights highlighting the potential of providing practical programming skills to FLOSS participants, have paved a way for a possible new education paradigm. This paradigm suggests incorporating participation in FLOSS projects as a requirement for some software engineering courses [8, 12–14, 25, 26]. A number of pilot studies have been conducted in order to evaluate the effectiveness of such an approach in traditional settings of learning [3, 12, 14, 22, 25, 26].

On the basis of such developments, an additional number of experiments have been conducted to provide empirical evidence with regards to how learning really occurs in FLOSS environments [6, 7, 16, 20]. This evidence suggests that the learning process in FLOSS environments occurs in 3 phases: Initiation, Progression and Maturation [18, 19, 21]. A description of these phases is provided through modeling and process maps using process mining [18, 20].

Although such a description is critical in laying the foundational ingredients, we believe that a more visual representation depicting learning activities and patterns can solidify evidence for the existence of learning opportunities in FLOSS communities. Therefore, in this paper we make use of the data and process visualisation technique in process mining [23] in order to provide a temporal visual distribution of learning events in Openstack [9]. The significance of such an approach is twofold. First, it helps provide insights on the level of commitment form learning participants. Secondly, it provides insights with regards to the intensity of learning activities occuring in these environments.

The reminder of the paper is structured as follows: Section 2 gives a brief overview on FLOSS repositories as learning environments; in Section 3 we describe learning processes in FLOSS communities; Section 4 provides a short description of our dataset as well as the dotted chart used to visualize it. Section 5 provides the results of our experiements and in Section 6 we discuss the results and conclude our study.

2 FLOSS Repositories as Learning Environments

A lot can be said in this section as ascertained and evidenced by ongoing projects as well as research findings from [1, 2, 4, 6, 7, 10, 24]. We opt to consider an interesting aspect of findings within the context of the FLOSScom project [6, 7] to exemplify a case where FLOSS repositories have been proved to be learning environments.

In this project, the investigators specifically focused on the analysis of informal learning in these communities [7]. The results of this study are based on a survey where 350 participants were asked to fill a questionnaire on skills acquired in FLOSS. The results indicate that there are 12 levels (cohorts) of professional expertise that testified to have learned something in FLOSS. There is an indication of a learning curve among the young cohorts as their learning process is gradual and expanding through skills improvements for different categories of skill set. This learning curve progressively starts initially with social skills when they express their opinions and interact with each other [2]. Results also indicate that the semi-experienced participants significantly increase their skills like the first cohort but with a special mention in programming skills [6, 7, 15]. The cohort of experienced participants demonstrates similar skills improvements but they also excel in managerial and legal skills.

These results indicate that participants in FLOSS communities acquire skills differently because of their professional background and experience. More importantly, young community members with no professional experience agree that FLOSS communities provide an adequate learning environment that provides for an informal but very comprehensive curriculum capable of making them full-fledged computer programmers, coordinators and activists [15].

We thus emphasize that the skills acquisition for FLOSS community members occurs while participating and performing a number of activities taking place in FLOSS repositories. FLOSS repositories such as CVS, Bug reports, mailing

archives, Internet relay chats etc., contain all traces of participants' activities as they work in these environments. During this process, FLOSS participants can post questions to mailing or discussions forums, review submitted pieces of software, debug code and report bugs etc.

3 Learning Processes in FLOSS Environments

The bulk of reports on FLOSS members' profiling have found that FLOSS members in these communities hold different roles that define their responsibilities and participation in the community activities [6, 7, 11]. These include testers, debuggers, project managers, co-developers and the core developers that make up the core development team. Among these roles, project initiators and the core development team remain at the heart of any development project in the community. This is made up of a small number of developers while the rest of contributors, referred to as the enhanced team, perform additional tasks such as feature sugges-tions, testing and query handling [11]. Apart from FLOSS participants who play roles with direct impact on FLOSS project, we can also distinguish between passive and active users of FLOSS products. Passive users are observers whose only active role is the mere use of the products. Active users are members of the community who do not necessarily contribute to the project in terms of coding, but whose support is made through testing and bug reporting [6, 7, 11].

As highlighted by Aberdour [1], participants increase their involvement in the project through a process of role meritocracy. This implies that passive users could move from their state of passiveness to active users, bug reporters until they possibly become part of the core team [20]. All these roles represent crucial contributions required for the overall project quality. However, in FLOSS environments, moving to a higher state is regarded as a reward and recognition of members' abilities and contributions [1]. Addionally, such role migration is also seen as moving to a higher skill level [10] exemplifying how new skills are developed in these environments.

Hence, it has been proposed that a typical learning process in FLOSS occurs in three main phases: Initiation, Progression and Maturation [18, 21]. In every phase, a number of activities are executed between Novices and Experts. A Novice is considered as any participant in quest of knowledge while the knowledge provider is referred to as the Expert [15]. Due to constraints related to space limitations in this paper, we illustrate only the initiation phase as depicted in Figures 1 and 2.

Principal activities gravitate around observing and making contacts in the Initiation Phase of the learning process [18]. Ideally, this step constitutes an opportunity for the Novice to ask questions and get some help depending on the requests while the Expert intervenes at this point to respond to such requests. On the one hand, when a Novice seeks help, he/she can execute a number of activities. These include FormulateQuestion, IdentifyExpert, PostQuestion, CommentPost or PostMessage, ContactExpert and SendDetailedRequest.

On the other hand, the main activities as undertaken by the Expert during the same period of time include ReadMessages on the mailing lists/Chat messages, ReadPost from forums, ReadSourceCode as any participant commits code to the project, or CommentPost, ContactNovice and CommentPost.

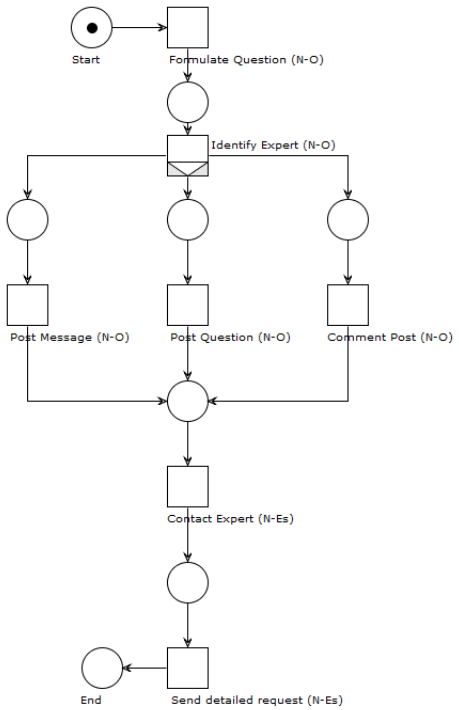

Fig. 1: Learning Process Model for Novice in Initiation Phase

4 Visualization of Openstack Learning Event Data with the Dotted Chart

The FLOSS platform used in this analysis is OpenStack [9]. According to Wikipedia, "OpenStack is a free and open-source software cloud computing software platform. Users primarily deploy it as an infrastructure as a service (IaaS) solution. The technology consists of a series of interrelated projects that control pools of pro-cessing, storage, and networking resources throughout a data center—which users manage through a web-based dashboard, command-line tools, or a RESTful API that is released under the terms of the Apache License" [27].

We considered this platform mainly due to the availability of data needed for our analysis and also because it is still an active platform. This database is

made up of 7 tables that store data pertaining to compressed files (source_code file, bugs), the mailing lists as per group discussions and topic of interests, the number of messages exchanged as well as details of the individuals involved in these exchanges. This repository contains exactly 54762 emails exchanged between 3117 people who are registered on 15 different mailing lists. These emails were sent during a period of time spanning from 2010 to 2014. The first message recorded (the very first email sent) was at 10:34:23 on the 11th of November 2010 while the last email considered was sent at 12:16:22 on the 6th of May 2014. The length of the messages considered is of typical email length specifically with an average of 3261 characters, the longest email was of 65535 characters and the shortest message yields a single character length [15]. This dataset (mailing archive) will be used to look at the first and second phases of the learning process.

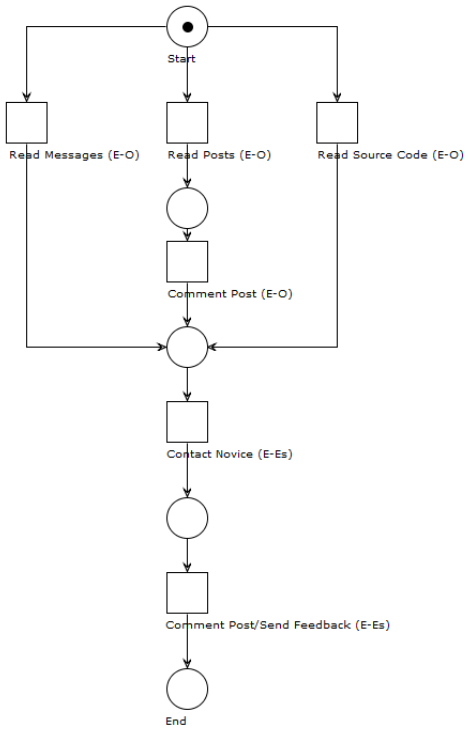

Fig. 2: Learning Process Model in Initiation Phase

In order to obtain a complete analysis for the third phase of the learning process, we will consider the code repository. This dataset contains information directly pertaining to commits, submitted codes as well as description and comments on the submitted code. This database is made up of 13 tables that we have considered. These tables contain data about the actions undertaken by de-

velopers, the pieces of codes (branches) as they commit them, the commit lines, files, their copies, the domains appropriate for these commits, the tags and their revisions, the projects as well as the people involved in these actions [15]. This repository contains exactly 93584 source code files that are reported to be committed. This is achieved by 2677 people. These people performed a number of actions and these among to 425744 on about 210 projects as reported. The files submitted are of 75441 types. This piece of information was particularly used to identify whether a file committed was documentation, user manual or patch or even any general source file. These bugs were reported during a period of time spanning from 2010 to 2014. About 131556 messages can be identified as they pertain to the committed files. The first message recorded was at 23:05:26 on the 27th of May 2010 while the last message included in this analysis was sent at exactly at 12:27:48 on the 6th of May 2014 [15].

The dotted chart is a discovery technique in proces mining [23] that provides a graphically representation of a process as it occurs over time. The chart enables the user to get invaluable insights pertaining to how events have occurred in relation to each other over the process lifespan. Providing a helicopeter view of the events data in a process, it is an interesting technique that gives critical hints pertaining to the performance of a process based on the time requirement [23]. A dotted chart, much like a Gantt chart, plots event data over time. On the chart, every dot represents a single event in a process occuring at a specific time. It has two orthogonal dimensions: a time and compoent dimensions [23]. The time is measured along the horizontal axis, while on the vertical axis, any component (instance, originator, case if, etc) pertaining to an event is represented.

In general, with such a temporal visual representation, one can identify variations in terms of duration in the way certain events occur etc. In our case, such a spread of learning events is crucial in providing insights regarding both the level of commitment of learning participants and the intensity of the learning cycle.

5 Results

We make use of the latest version of the dotted chart to visualize data about learning processes. Although we, already performed a series of analysis on the same dataset in [15, 20, 21], this additional perspective is critical as it not only enriches evidenciary learning considerations in FLOSS communities, it also simplifies the representation of learning patterns in these communities. We set to look at two specific things: the level of commitment learning participants exhibit throughout the process lifespan and the overall intensity related to learning therein.

These two factors are critical in light of current trends for learning behaviours in Massive Open Online Course (MOOCs). Analysis in such environments has demonstrated that students usually start strong and that the level of commitment decreases over time [17] [16]. This trend is a typical characteristic of the level of commitment from students, which eventually expresses their motivation

and interests. Hence, the results of such an analysis in FLOSS would provide interesting observations that are crucial to the value and reputation of FLOSS environments as learning environments.

We set our parameters (dimensions) as the participants (active people performing activities) versus the time at which they performed those activities. Hence, the horizontal axis represents the time point at which learning participants on the vertical axis executed a learning activity. The resulting dotted charts are given in Figures 3, 4, 5, 6, 7, and 8. The figures (visualizations) respectively represent the learning trends for both the Novice and Expert in the Initiation, Progression and Maturation of the learning process.

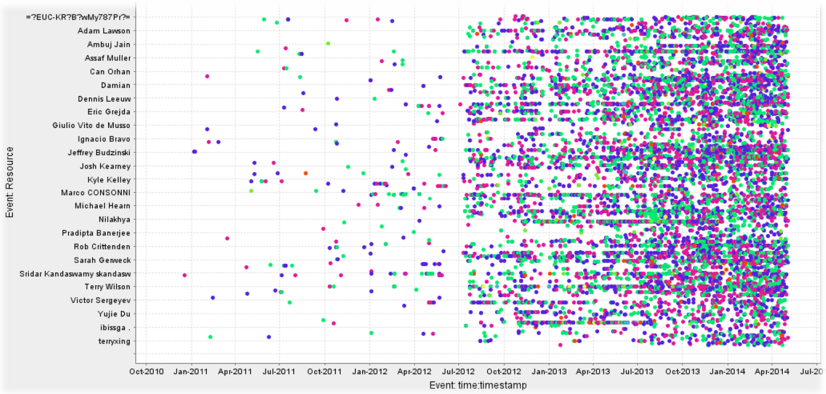

Fig. 3: Temporal Visualization of Novice's Learning Activities during Initiation Phase on Mailing Archives where each dot represents a learning activity

6 Discussion and Conclusion

A number of studies on FLOSS environments have laid a foundation regarding the potential for the existence of learning opportunities in these communities. However, we believe that more could be done in terms of providing additional empirical evidence and visualizations for learning processes in these environments.

Therefore, in this paper we set to make use of a powerful process mining discovery technique called dotted chart in order to provide a temporal visual representation of activities and learning patterns as they occur over time. We made use of the datasets provided through the Openstack platform and we sought to verify two important factors. Firstly, we wanted to get some indications pertaining to the level of commitment from learning participants throughout the learning process. Secondly, we also wanted to get insights with regards to the intensity of learning activities occurring in these environments. The motivation for

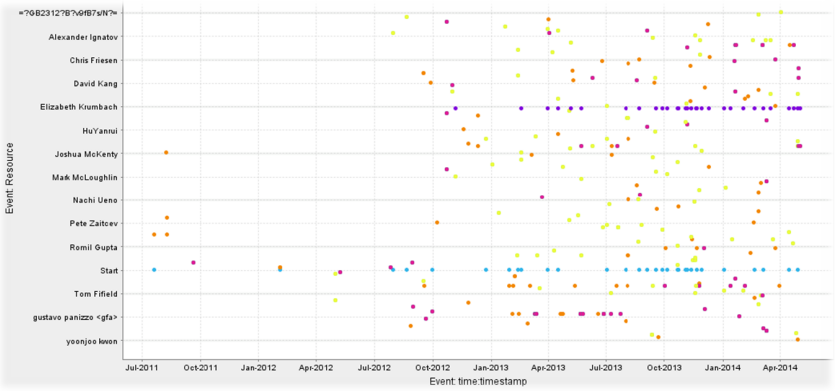

Fig. 4: Temporal Visualization of Expert's Learning Activities during Initiation Phase on Mailing Archives where each dot represents a learning activity

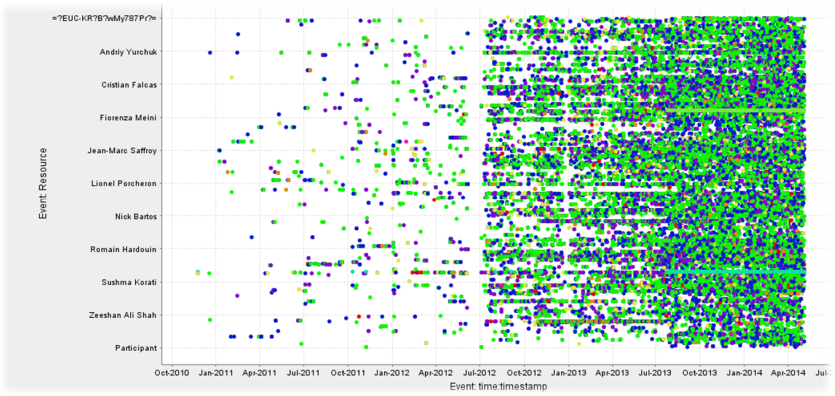

Fig. 5: Temporal Visualization of Novice's Learning Activities during Progression Phase on Mailing Archives where each dot represents a learning activity

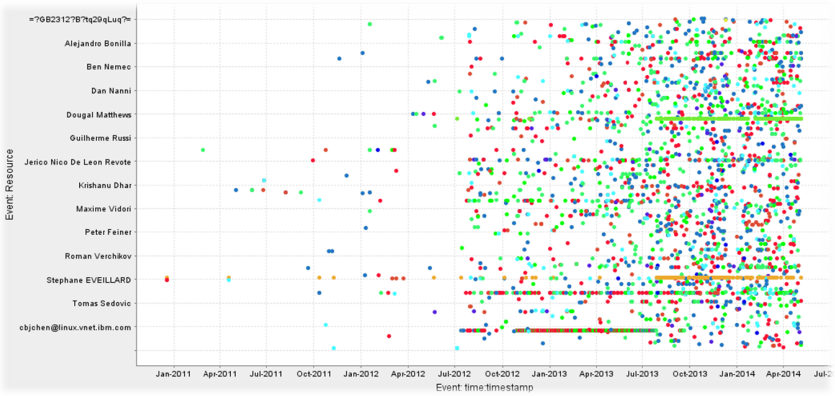

Fig. 6: Temporal Visualization of Expert's Learning Activities during Progression Phase on Mailing Archives where each dot represents a learning activity

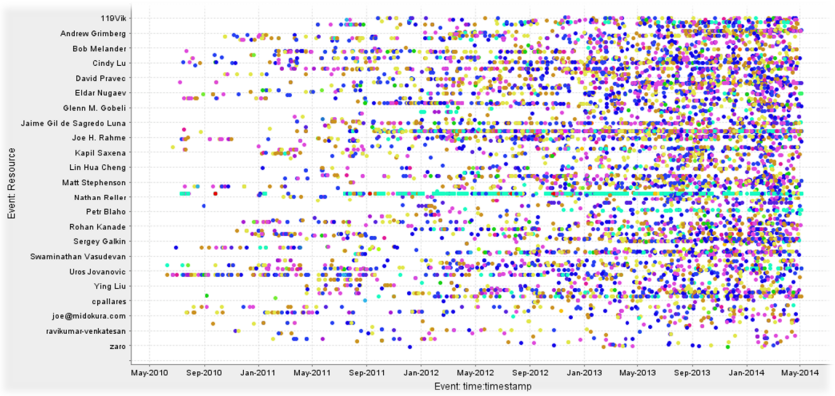

Fig. 7: Temporal Visualization of Novice's Learning Activities during Maturation Phase on Mailing Archives where each dot represents a learning activity

Fig. 8: Temporal Visualization of Expert's Learning Activities during Maturation Phase on Mailing Archives where each dot represents a learning activity

an analysis predicated on these factors is inspired by the current trends on learning behaviours in MOOCs. Studies suggest that in MOOCs, only a very small percentage of students complete the course in spite of having a large number of students enrolling and performing some activities in the first days.

Our results indicate that, in contrast to the trends in MOOCs, learning is sustainable in FLOSS environments. Looking at Figures 3, 4, 5, 6, 7, and 8, we can almost consistently notice that as the learning participants (Novice and Expert) engage in the learning process, instead of reducing their participation in learning activities, on the contrary, the level of commitment increases. One can notice that as the Novice acquires new knowledge, the learning atmosphere is intensified as evidenced by the multitude of events in which the Novice is involved in accross the 3 phases.

We can, therefore, say that FLOSS environments provide sustainable and consistent opportunities for people to learn new skills and this occurrs pretty consistently from the first to the last phase. Our results provide interesting insights that lay the ground for future exploration and introspection into analysis of the level of intensity and learning participation in FLOSS environments.

References

1. Aberdour, M.: Achieving quality in open source software. IEEE Software 24(1), 58–64 (2007), http://dx.doi.org/10.1109/MS.2007.2
2. Cerone, A.: Learning and activity patterns in OSS communities and their impact on software quality. ECEASST 48 (2011), http://journal.ub.tu-berlin.de/eceasst/article/view/803
3. Cerone, A., Sowe, S.K.: Using free/libre open source software projects as e-learning tools. Electronic Communications of the EASST 33 (2010)
4. Dillon, T., Bacon, S.: The potential of open source approaches for education. FutureLab Opening Education Reports, http://www. futurelab. org.

uk/resources/publicationsreports-articles/opening-education-reports/Opening-Education-Report200 (2006)
5. Fernandes, S., Barbosa, L.S., Cerone, A.: Floss communities as learning networks. International Journal of Information and Education Technology 3(2), 278 (2013)
6. Glott, R., Meiszner, A., Sowe, S.: Flosscom phase 1 report: analysis of the informal learning environment of floss communities. FLOSSCom Project (2007)
7. Glott, R., SPI, A.M., Sowe, S.K., Conolly, T., Healy, A., Ghosh, R., Karoulis, A., SPI, H.M., Stamelos, I., Weller, M.J., et al.: Flosscom-using the principles of informal learning environments of floss communities to improve ict supported formal education (2011)
8. Jaccheri, L., Østerlie, T.: Open source software: A source of possibilities for software engineering education and empirical software engineering. In: Emerging Trends in FLOSS Research and Development, 2007. FLOSS'07. First International Workshop on. pp. 5–5. IEEE (2007)
9. Jackson, K., Bunch, C., Sigler, E.: OpenStack cloud computing cookbook. Packt Publishing Ltd (2015)
10. Jensen, C., Scacchi, W.: Role migration and advancement processes in OSSD projects: A comparative case study. In: 29th International Conference on Software Engineering (ICSE 2007), Minneapolis, MN, USA, May 20-26, 2007 [10], pp. 364–374, http://dx.doi.org/10.1109/ICSE.2007.74
11. Krishnamurthy, S.: Cave or community? an empirical examination of 100 mature open source projects. First Monday 7(6) (2002), http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/960
12. LeBlanc, R.J., Sobel, A., Diaz-Herrera, J.L., Hilburn, T.B., et al.: Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. IEEE Computer Society (2006)
13. Meiszner, A., Glott, R., Sowe, S.K.: Free/libre open source software (floss) communities as an example of successful open participatory learning ecosystems. UPGRADE, The European Journal for the Informatics Professional 9(3), 62–68 (2008)
14. Meiszner, A., Mostaka, K., Syamelos, I.: A hybrid approach to computer science education–a case study: software engineering at aristotle university (2009)
15. MUKALA, M.P.: Process Models for Learning Patterns in FLOSS Repositories. Ph.D. thesis (2015)
16. Mukala, P., Buijs, J.C.A.M., van der Aalst, W.M.P.: Exploring students' learning behaviour in moocs using process mining techniques. Tech. rep., Eindhoven University of Technology, BPM Center Report BPM-15-10, BPMcenter.org (2015)
17. Mukala, P., Buijs, J.C.A.M., Leemans, M., van der Aalst, W.M.P.: Learning analytics on coursera event data: A process mining approach. In: Proceedings of the 5th International Symposium on Data-driven Process Discovery and Analysis (SIMPDA 2015), Vienna, Austria, December 9-11, 2015. pp. 18–32 (2015), http://ceur-ws.org/Vol-1527/paper2.pdf
18. Mukala, P., Cerone, A., Turini, F.: An abstract state machine (ASM) representation of learning process in FLOSS communities. In: Software Engineering and Formal Methods - SEFM 2014 Collocated Workshops: HOFM, SAFOME, OpenCert, MoKMaSD, WS-FMDS, Grenoble, France, September 1-2, 2014, Revised Selected Papers [18], pp. 227–242, http://dx.doi.org/10.1007/978-3-319-15201-1_15
19. Mukala, P., Cerone, A., Turini, F.: Ontolifloss: Ontology for learning processes in FLOSS communities. In: Software Engineering and Formal Methods - SEFM 2014 Collocated Workshops: HOFM, SAFOME, OpenCert, MoKMaSD, WS-FMDS, Grenoble, France, September 1-2, 2014, Revised Selected Papers [19], pp. 164–181, http://dx.doi.org/10.1007/978-3-319-15201-1_11

20. Mukala, P., Cerone, A., Turini, F.: An exploration of learning processes as process maps in floss repositories (2015)
21. Mukala, P., Cerone, A., Turini, F.: Mining learning processes from FLOSS mailing archives. In: Open and Big Data Management and Innovation - 14th IFIP WG 6.11 Conference on e-Business, e-Services, and e-Society, I3E 2015, Delft, The Netherlands, October 13-15, 2015, Proceedings [21], pp. 287–298, http://dx.doi.org/10.1007/978-3-319-25013-7_23
22. Papadopoulos, P.M., Stamelos, I., Meiszner, A.: Enhancing software engineering education through open source projects: Four years of students' perspectives. EAIT 18(2), 381–397 (2013), http://dx.doi.org/10.1007/s10639-012-9239-3
23. Song, M., van der Aalst, W.M.: Supporting process mining by showing events at a glance. In: Proceedings of the 17th Annual Workshop on Information Technologies and Systems (WITS). pp. 139–145 (2007)
24. Sowe, S.K., Stamelos, I.: Reflection on knowledge sharing in F/OSS projects. In: Open Source Development, Communities and Quality, IFIP 20th World Computer Congress, Working Group 2.3 on Open Source Software, OSS 2008, September 7-10, 2008, Milano, Italy [24], pp. 351–358, http://dx.doi.org/10.1007/978-0-387-09684-1_32
25. Sowe, S.K., Stamelos, I., Deligiannis, I.S.: A framework for teaching software testing using F/OSS methodology. In: Open Source Systems, IFIP Working Group 2.13 Foundation on Open Source Software, June 8-10, 2006, Como, Italy [25], pp. 261–266, http://dx.doi.org/10.1007/0-387-34226-5_26
26. Sowe, S.K., Stamelos, I.G.: Involving software engineering students in open source software projects: Experiences from a pilot study. Journal of Information Systems Education 18(4), 425 (2007)
27. Wikipedia: Openstack — wikipedia, the free encyclopedia (2016), https://en.wikipedia.org/w/index.php?title=OpenStack&oldid=716661364, [Online; accessed 24-April-2016]