Mining Educational Social Network Structures from FLOSS Repositories

Patrick Mukala, Ph.D

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

Abstract. FLOSS environments have been proved to provide an interesting learning platform for software engineers. Research suggests that people partaking in both technical and non-technical activities in FLOSS prjects are more likely to positively improve their software engineering skills. To this end, there are propositions to involve computer science and software engineering students in formal higher institutions of learning, in participating in FLOSS projects in order to give them an opportunity to develop their programming capacity by working on real-life projects. While some empirical studies have been conducted to provide some lights on learning processes in FLOSS environments, there is limited or no work done pertaining to understanding social structures during this process of knowledge transfer and acquisition. In this paper, we make use of social network analysis techniques in order to provide insights related to the emerging of social structures from FLOSS repositories from an educational point of view. We hope that these educational structures will enhance both the understanding with regards to how learning occurs in these communities and especially, the frequency of participants' involvement that culminates into learning.

Keywords: Social Network Analysis, Learning Models, Mining Data, Educational Data Mining, Learning in FLOSS, Online Learning, Openstack

1 Introduction

Recent studies have raised the level of interests in studying the existence of learning opportunities and knowledge exchange in Free/Libre Open Source Software (FLOSS) environments [6–9,12,17,29,31].Some of these studies establish FLOSS communities as environments where successful collaborative and participatory learning between participants occurs [7, 9, 17]. 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 [12, 16–18, 30, 31]. A number of pilot studies have been conducted in order to evaluate the effectiveness of such an approach in traditional settings of learning [7, 16, 18, 26, 30, 31].

On the basis of such developments, an additional number of experiments have been conducted to provide some empirical evidence in this direction [10,11,21,24]. This evidence suggests that the learning process in FLOSS environments occurs in 3 phases: Initiation, Progression and Maturation [22, 23, 25]. A description of these phases is provided through modeling and process maps using process mining [22, 24].

While the current work in this area is critical in supporting the empirical evidence for the existence of learning opportunities in FLOSS communities, we propose to further contribute by mining social structures in these environments. Social structures or networks are critical in understanding collaboration patterns between people who are involved in common activities [2, 5, 32]. The benefits of social networks are varied and at a glance, one can note that they provide a deep understanding into interaction patterns and clustering from data, which can be leveraged to make informed decisions in a number of contexts [2, 4, 5, 19, 27]. Within the realm of FLOSS environments, there has been a study to understand how people interact in general and how they can move from one repository to another [5]. Therefore, in this paper we set to explore social structures or formations that take place while people learn in FLOSS environments.

The purpose of such an exploration is to provide on-the-fly exploration of social structures that can explain the level of commitment and learning intensity exhibited by learning participants in FLOSS communities. Since we are concerned with learning processes, we will make use of the tool in ProM [1].

The reminder of the paper is structured as follows. In Section 2 we give summarised description of learning processes in FLOSS environments. Section 3 details our dataset for the context of this analysis. In Section 4 we describe the results our experiments and in Section 6 we discuss the results and conclude our study.

2 Learning Processes in FLOSS Environments

The learning processes in these environments simply encompass the paths participants take while performing a number of activities in FLOSS repositories. FLOSS repositories such as CVS, Bug reports, mailing archives, Internet relay chats etc., contain all traces of participants' activities as ascertained and evidenced by ongoing projects as well as research findings from [3, 6, 8, 10, 11, 14, 29].

The basis of our definition of these learning processes stems from observations in the bulk of reports on FLOSS members' profiling [10,11,15]. The reports or studies have found that FLOSS members in these communities hold different roles that define their responsibilities and participation in the community activities [10,11,15]. 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 suggestions, testing and query han-

dling [15]. 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 [10, 11, 15].

As highlighted by Aberdour [3], 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 [24]. 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 [3]. Addionally, such role migration is also seen as moving to a higher skill level [14] 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 [22, 25]. 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 [22, 25] [20]. Due to constraints related to space limitations in this paper, we illustrate only the Progression phase as depicted in Figures 1 and 2.

To quickly summarise the progression phase, one should note that in this phase both Novice and Expert execute a series of new activities building up from the Initiation phase phase. As depicted in Figures Figure 1 and 2. After accepting a request from the Novice, the Expert performs ReviewThreadPosts to be fully aware of the questions and needs for clarification raised by the Novice and ReviewThreadCode, for the purpose of critiquing and fixing the code, if needed. The Expert may also perform SendReply in an attempt to answer any direct questions and help requests or just to react to a discussion in a forum. Furthermore, the Expert performs SendFeedback and ReplyPostedQuestion, to directly or indirectly address doubts or questions from the Novice, PostQuestions, to enquiry about possible further needs by the Novice, and ReportBugs, as a response to Novice's needs, such as understanding why a piece of code does not run properly.

Moreover, the Expert may monitor the Novice through a set of activities for the purpose of evaluating the level of skill acquisition. These activities include RunSourceCode and AnalyseSourceCode, to identify flaws in the Novice's works, and, if necessary, ReportBugs, CommentOnCode and ReplyToPost [22, 23].

The Novice can only react to the Expert's help or feedback by providing insights on the extent to which such help or feedback was useful through ProvideFeedback, or simply posing more questions through PostQuestions. The Novice also performs a number of activities in the context of posting. These activities may include PostQuestions, ReplyPostedQuestions and possibly SendFeedback.

Furthermore, the Novice can start exercising the new acquired skills through activities such as AnalyseSourceCode, when looking at new commits, new pieces of code being posted by community members. Thus, the Novice is able to comment on commits and code through CommentOnCode and by reporting bugs through ReportBugs [20, 22].

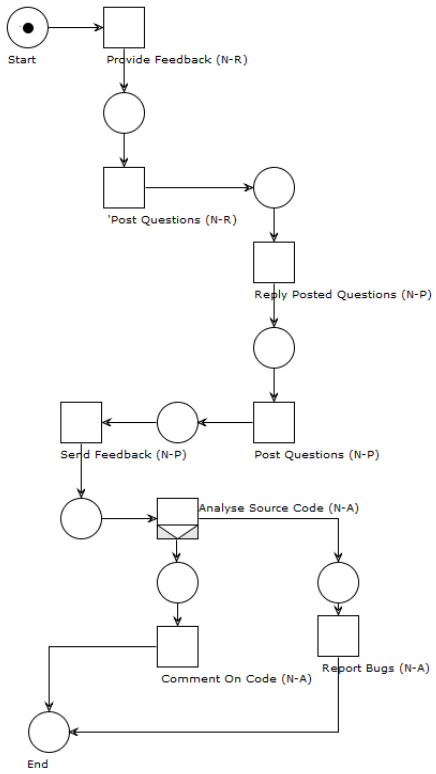


Fig. 1: Learning Process Model for Novice in Initiation Phase

3 Educational Social Structures from Openstack Learning Event Data

The FLOSS platform used in this analysis is OpenStack [13]. According to Wikipedia, "OpenStack is a free and open-source software cloud computing software plat-form. 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 cen-

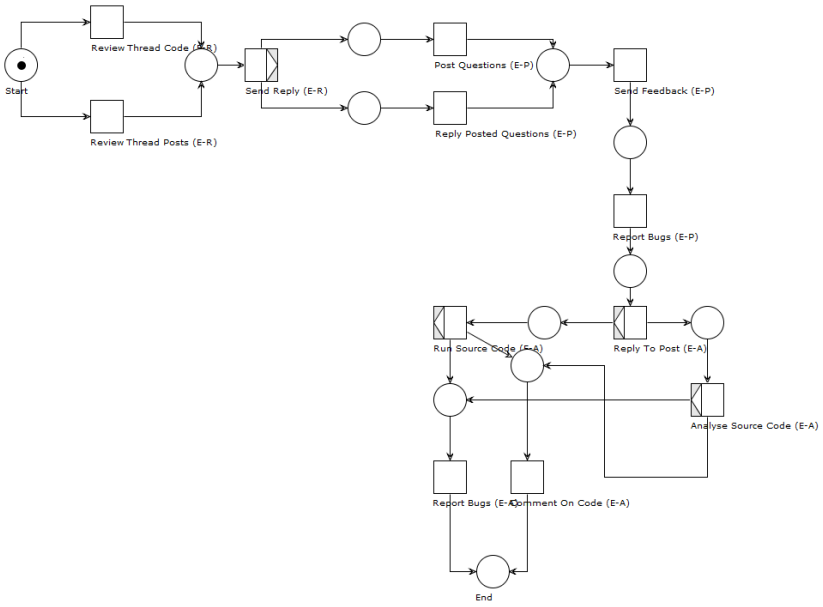


Fig. 2: Learning Process Model in Initiation Phase

ter—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" [33].

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 [20] [25].

This dataset is convenient for our analysis as activities from both the Initiation and Progression phases can be traced and mined on mailing archives. For a quick glance of the data and learning activities in this phase, we made use of the dotted chart as depicted in Figures 3, 4.

The dotted chart is a discovery technique in proces mining [28] 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 proces, it is an interesting technique that gives critical hints peratining to the performance of a process based on the time requirement [28]. 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 [28]. 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.

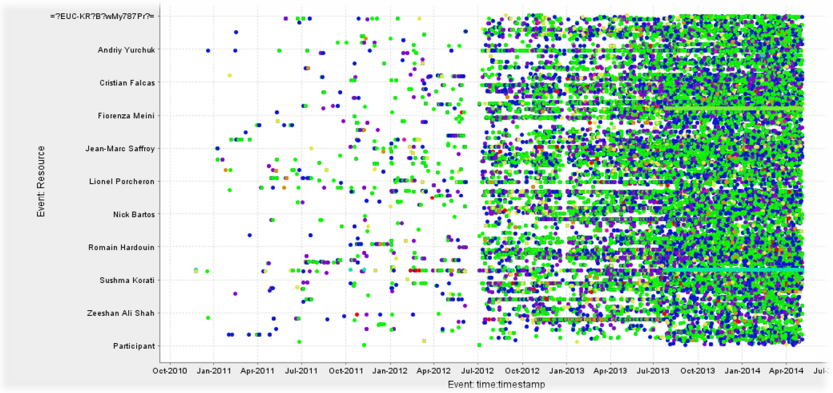


Fig. 3: Temporal Visualization of Novice's Learning Activities during Progression Phase on Mailing Archives

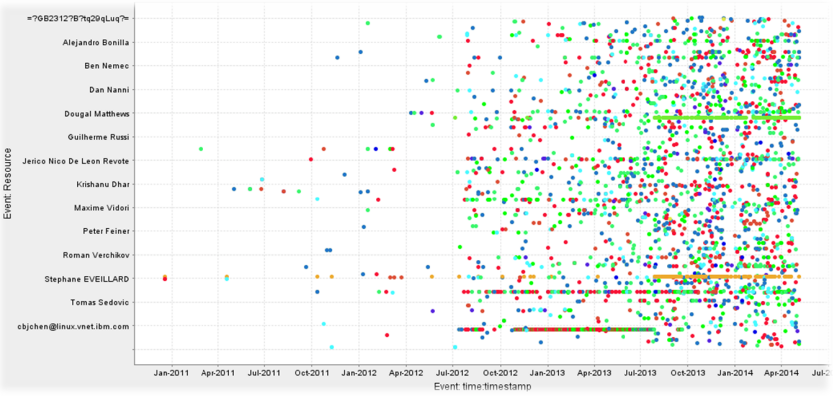


Fig. 4: Temporal Visualization of Expert's Learning Activities during Progression Phase on Mailing Archives

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 the learning cycle. We set our parameters (dimensions) as the participants (active people performing activities) versus the time at which they performed those activities. Figures 3, 4 demonstrate that learning activities occur consistently in our FLOSS environment and increase as participants kove from the Inition to the Progression phase.

We have a general picture of learning event data at this point given through our dotted charts. We therefore focus on uncovering collaboration patterns in order to uncover roles and entities defining the relation between people or groups of people that are interacting and the process. Another perspective is to focus on the relations among individuals (or groups of individuals) acting in the process [32].

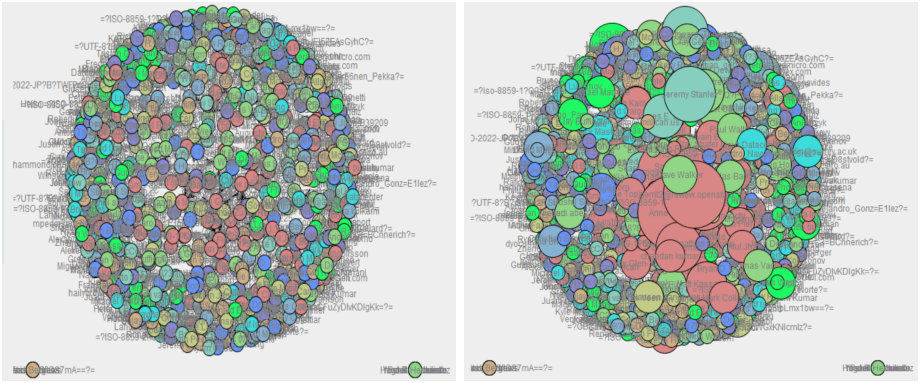
In order to conduct our analysis, we make use of the MiSoN tool implemented in ProM [1]. In this paper, we do not undertake a formal and semantic analysis of social networks metrics/properties such as centrality, betweeness, closeness on our data, we rather measure the level of learning on mailing archives from a social network perspective.

4 Results

On extracting social structures from our learning event logs, we can consider a range of metrics including *subcontracting*, *hand over of work (transfer of work), working together task and similar task [2, 32].*

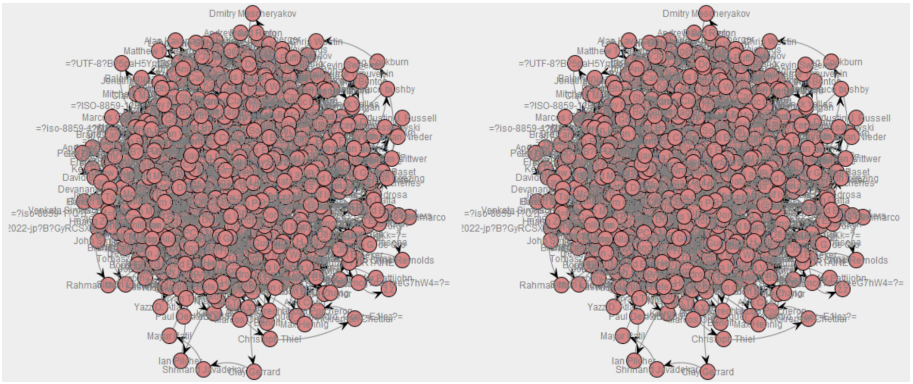
There is a *hand over of work* from individual i to individual j if there are two subsequent activities where the first is completed by i and the second by j. This is an intersing pattern that can reveal the degree at which learning participants in FLOSS environments intervene in the learning process by performing certain random activities at any given phase. For the *subcontracting* metric which is related to the transfer of work metric, the main idea is to count the number of times individual j executed an activity in-between two activities executed by individual i. With the *working together metric*, the idea is simply to count how frequently two individuals are performing activities for the same case. In our analysis, this will provide insights on the enthousiasm exhibited by learning participants acquiring or transfering a skill. The assumption with the last metric, *similar task,* is that people executing the similar activities have stronger relations than people executing completely different activities. Each individual has a "profile" based on how frequent they conduct specific activities [2, 32].

We only consider the first 3 metrics to extract our social networks depicting learning activities as seen in Figures 5, 6 and 7 for the Novice and Figures 8, 9 and 10 for the Expert.



(a) Generic Social Network (b) Nodes sized and ranked by their degree

Fig. 5: Social Network for Novice following the subcontracting metric



(a) Generic Social Network (b) Nodes sized and ranked by their degree

Fig. 6: Social Network for Novice following the handover of work metric

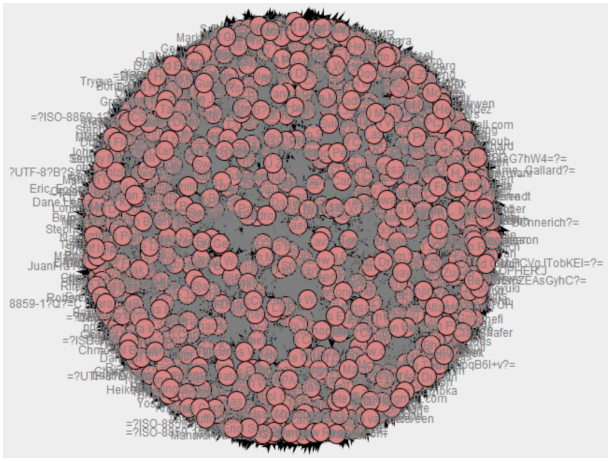
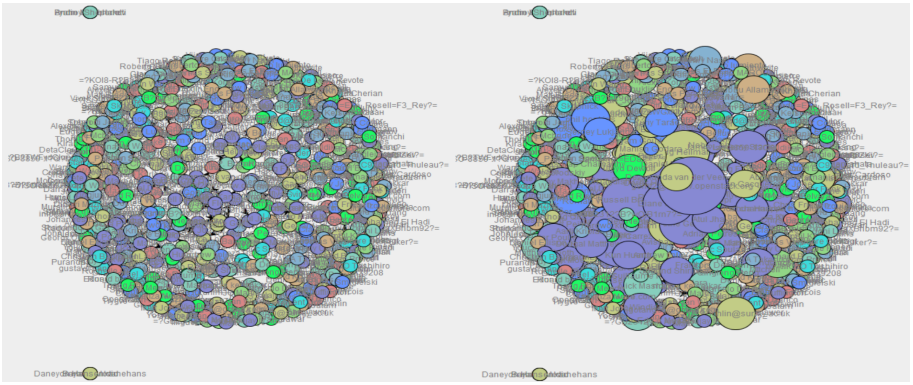
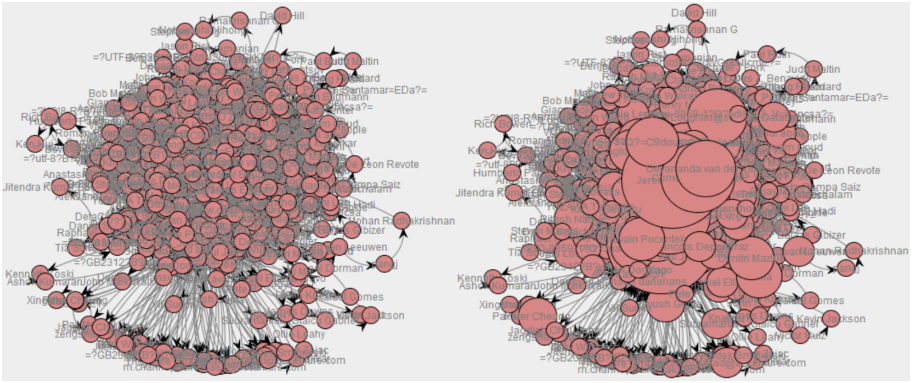


Fig. 7: Social Network for Novice following the handover of work metric



(a) Generic Social Network (b) Nodes sized and ranked by their degree

Fig. 8: Social Network for Expert following the subcontracting metric



(a) Generic Social Network (b) Nodes sized and ranked by their degree

Fig. 9: Social Network for Expert following the handover of work metric

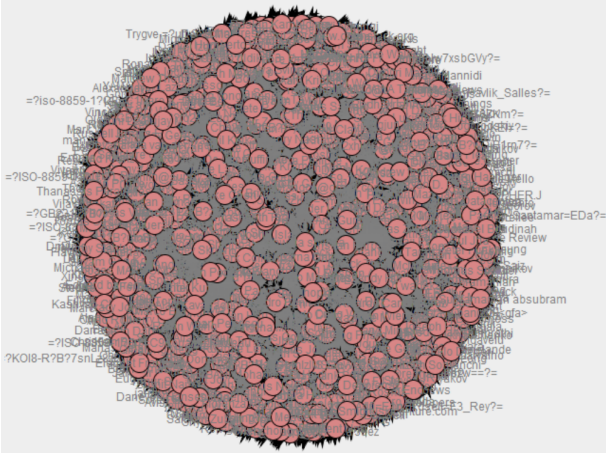


Fig. 10: Social Network for Expert following the handover of work metric

5 Discussion and Conclusion

A number of studies on FLOSS environments have laid a foundation regarding the potential for the occurrence of learning in these communities. As such, it is critical to analyze and get insights with regards to learning behaviour exhibited by learning participants.

With the hope to provide more empirical evidence and visualizations for learning processes in FLOSS environments, we set to make use of social analysis metrics to this end. In this paper, we focused on deriving social structures from event logs based on 3 specif metrics : *subcontracting*, *hand over of work (transfer of work)*, *working together task*.

The results of our analysis demonstrate a high-level of commitment in the learning process on Mailing archives. Particularly in this progression phase, Figures 5, 6 and 7 for the Novice and Figures 8, 9 and 10 for the Expert, showcase at what extent people are constantly engaged in the knowledge generation and exchange. A lot of traffic is generated as an activity is executed, multiple participants engage in these activities at any point and only a few (2) outliers can be notice accross the 3 metrics.

In future, we plan to extend this analysis with a thoughrough detailing per activity to drill down and understand how these metrics explain the learning behaviour based on every single activity.

References

1. van der Aalst, W.M.P.: Process Mining - Discovery, Conformance and Enhancement of Business Processes. Springer (2011)
2. Van der Aalst, W.M., Song, M.: Mining social networks: Uncovering interaction patterns in business processes. In: Business Process Management, pp. 244–260. Springer (2004)
3. Aberdour, M.: Achieving quality in open source software. IEEE Software 24(1), 58–64 (2007), `http://dx.doi.org/10.1109/MS.2007.2`
4. Berkman, L.F., Syme, S.L.: Social networks, host resistance, and mortality: a nine-year follow-up study of alameda county residents. American journal of Epidemiology 109(2), 186–204 (1979)
5. Bird, C., Gourley, A., Devanbu, P., Gertz, M., Swaminathan, A.: Mining email social networks. In: Proceedings of the 2006 international workshop on Mining software repositories. pp. 137–143. ACM (2006)
6. 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`
7. Cerone, A., Sowe, S.K.: Using free/libre open source software projects as e-learning tools. Electronic Communications of the EASST 33 (2010)
8. 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)
9. Fernandes, S., Barbosa, L.S., Cerone, A.: Floss communities as learning networks. International Journal of Information and Education Technology 3(2), 278 (2013)

10. Glott, R., Meiszner, A., Sowe, S.: Flosscom phase 1 report: analysis of the informal learning environment of floss communities. FLOSSCom Project (2007)
11. 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)
12. 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)
13. Jackson, K., Bunch, C., Sigler, E.: OpenStack cloud computing cookbook. Packt Publishing Ltd (2015)
14. 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 [14], pp. 364–374, http://dx.doi.org/10.1109/ICSE.2007.74
15. 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
16. 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)
17. 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)
18. Meiszner, A., Mostaka, K., Syamelos, I.: A hybrid approach to computer science education–a case study: software engineering at aristotle university (2009)
19. Mislove, A., Marcon, M., Gummadi, K.P., Druschel, P., Bhattacharjee, B.: Measurement and analysis of online social networks. In: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement. pp. 29–42. ACM (2007)
20. MUKALA, M.P.: Process Models for Learning Patterns in FLOSS Repositories. Ph.D. thesis (2015)
21. 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)
22. 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 [22], pp. 227–242, http://dx.doi.org/10.1007/978-3-319-15201-1_15
23. 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 [23], pp. 164–181, http://dx.doi.org/10.1007/978-3-319-15201-1_11
24. Mukala, P., Cerone, A., Turini, F.: An exploration of learning processes as process maps in floss repositories (2015)
25. 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 [25], pp. 287–298, http://dx.doi.org/10.1007/978-3-319-25013-7_23
26. 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
27. Snow, D.A., Zurcher Jr, L.A., Ekland-Olson, S.: Social networks and social movements: A microstructural approach to differential recruitment. American sociological review pp. 787–801 (1980)
28. 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)
29. 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 [29], pp. 351–358, http://dx.doi.org/10.1007/978-0-387-09684-1_32
30. 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 [30], pp. 261–266, http://dx.doi.org/10.1007/0-387-34226-5_26
31. 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)
32. Van Der Aalst, W.M., Reijers, H.A., Song, M.: Discovering social networks from event logs. Computer Supported Cooperative Work (CSCW) 14(6), 549–593 (2005)
33. Wikipedia: Openstack — wikipedia, the free encyclopedia (2016), https://en.wikipedia.org/w/index.php?title=OpenStack&oldid=716661364, [Online; accessed 24-April-2016]