FLOSS Communities as Learning Networks

Sara Fernandes, Luis S. Barbosa, and Antonio Cerone

***Abstract*—This paper discusses FLOSS communities as a prime example of learning networks, i.e. informal, interconnected sets of individuals mutually supporting shared knowledge acquisition and committed to self-defined goals. In particular, it addresses how the incorporation of a web-based, certification platform for Free / Libre Open Source Software in community culture and practice may increase the community impact both as learning network and open, high-quality software producer.**

***Index Terms*—FLOSS, communities, learning networks, source code analysis, educational technology.**

I. Introduction

Current trends in education point out that learning result from participation in social interactions and in culturally organized activities with others [1]. This shift of perspective raises a number of questions on the organization of the educational process, its dynamics to provide suitable support with different degrees of formality.

E-learning systems and e-learning supported infrastructures are certainly part of this debate. It has been pointed out [2] that, "in the last 20 years, e-learning grew from a unique college experiment to a full category of higher education. In 2010, there were more people enrolled in online classes than the entire population of Wisconsin". It is therefore legitimate to think of a fully personalized education system designed around needs, interests and aspirations of each learner.

Moreover, with the emergence of Web 2.0, conventional e-Learning systems [3], based on instructional packets and cumulative assignments, gives the stage to a different reality which promotes the concept of social learning through the use of social software tools, such as blogs, wikis, forums, etc. As a result, learning in a broad and heterogeneous perspective, occurs at a societal level through the development of complex interactions between peers [4].

This new learning perspective is in contrast with a number of fundamental assumptions which have historically underpinned the organization of education: 1) expertise and knowledge resides only within the walls of the educational institution; 2) "learning" and "schooling" are different words for the same thing; 3) the most "equitable" educational systems are those which offer a "one-size-fits-all" approach, and 4) the easiest and most cost-effective approach to organizing learning is within the walls of the school [5].

This paper tries to make a concrete contribution to re-thinking educational practices in computer-oriented environments. Far away from the "school as a factory" metaphor, we envisage learning approaches that, in sharp contrast with formal institutions and curricula, promote more porous and flexible learning networks linking homes, communities and multiple sites of learning.

Our focus is on a prime example, even if somewhat less conventional: the learning process that emerges from participation in developing communities for Free/Libre Open Source Software (FLOSS). FLOSS projects are a result of open and participatory development processes taking place within communities of practice. In general they offer an instantiation of peer-production models, aiming at articulating the creative power of large numbers of individuals, remotely coordinated, typically through the Internet, into large and complex projects, most of them without any sort of hierarchical organization [6]. Thus, FLOSS projects can be seen as collaborative learning environments developed inside communities in the precise sense of "collectives of individuals that cohere around a shared spirit" [7]. They are a prime example of what in the literature is referred to as a learning network. That is, a community of individuals developing and maintaining connections among them, sharing information, and communicating in such a way to support one another's learning [8].

In this context, the aim of this paper is twofold. First it aims at framing FLOSS communities as learning networks, whose potential deserves to be better explored. Second it discusses how rigorous code analysis techniques can be integrated in such communities in a way that is relevant both for developers (by expanding their own learning network) and end-users (who aim at certified FLOSS components for integration in large software installations).

The rest of the paper is structured as follows. Section 2 recalls the notion of learning network and discusses how FLOSS communities can be regarded as such networks. Section 3 focus on quality assurance in FLOSS products and introduces the CROSS portal, developed at Minho University, Portugal, as a web-based platform for supporting FLOSS analysis and certification. Section 4 formulates a working proposal for the integration of the CROSS portal in FLOSS communities and discusses its contribution to knowledge building and open software development. Finally, Section 5,

Manuscript received October 15, 2012; revised January 15, 2013. This work is funded by ERDF - European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness), by National Funds through the FCT - Fundacão para a Ciê ncia e a Tecnologia (Portuguese Foundation for Science and Technology) within project FCOMP-01-0124-FEDER-010047, as well as by UNU-IIST and Macau Foundation in the context of the PPAeL project.

S. Fernandes is with the United Nations University – International Institute for Software Technology, Casa Silva Mendes, Estrada Engenheiro Trigo NO 4, P.O. Box 3058 Macau (e-mail: sara.fernandes@ iist.unu.edu).

L. S. Barbosa is with Minho University and HASLab, INESC TEC, Campus de Gualtar, Braga 4710-400 Portugal.

A. Cerone is with the United Nations University – International Institute for Software Technology, Casa Silva Mendes, Estrada Engenheiro Trigo, No. 4, P.O. Box 3058 Macau (e-mail: cerone@iist.unu.edu).

DOI: 10.7763/IJIET.2013.V3.280

concludes and gives a number of pointer to current and future work.

II. FLOSS Communities as Learning Networks

The University of the First Age claims [9] that "deep learning flourishes where you build connecting relationships". Educational and social research increasingly gives room to new understandings of learning processes, acknowledging their often networked and collaborative properties. Moreover, Dirckinck-Holmfeld et al. claim "the network is now the fundamental underpinning structure of social organization – and that it is in and through networks – both real and virtual – that life is lived in the 21st century". Current education systems, exclusively based on individualized acquisition of content and skills, are, therefore, unsuitable to support learning processes and the development of individual and group competencies as an essentially networked ability.

Learning networks may arise both in informal and formal educational settings. A prime example of informal learning occurs in FLOSS communities, along project development. FLOSS communities consist of heterogeneous groups of independent volunteers, who interact driven by different motivations and with different backgrounds. Moreover, they act within a collaborative structure, based on open organization, interaction and knowledge exchange. FLOSS communities are essentially participatory: newcomers are gradually integrated along a process of skill acquisition (not only technical, but to a large extent, connected to distributed project development and management). Cerone and Sowe [10] distinguish four main, partially overlapping stages in the underlying learning process: i) socialise by implicit knowledge sharing; ii) externalise tacit knowledge by making it explicit to the community; iii) combine community explicit knowledge and re-organize it as abstract knowledge; iv) internalise such an abstract knowledge by absorbing it and combining it with own knowledge and experiences to produce new tacit knowledge.

The outcome of this process is far broader than just source code: a large variety of resources relevant both to the product (e.g. documentation) and the process (e.g., communication norms) are produced and made available to participants and the world outside.

Another aspect in FLOSS communities of practice is fundamental to build learning networks: the incremental combination of (formal and informal, explicit and implicit) knowledge. This is achieved through multiple interactions, mediated by knowledge-management tools, to identify such sources of knowledge and their combination therefore feeding the community.

In summary, FLOSS communities, by their own characteristics, are potentially true learning networks; communication and collaborative production being essential assets in that respect. In such a context, Sections III and IV below discuss how such a potential can be of use to both introduce rigorous analysis methods in FLOSS practice and integrate them in the shared culture of FLOSS communities.

III. Quality Analysis for FLOSS

Over the past decade, FLOSS phenomenon had a global impact on the way software systems and software-based services are developed, distributed and deployed. Widely acknowledged benefits of FLOSS include reliability, low development and maintenance costs, as well as rapid code turnover. Linux distributions, Apache and MySQL, serve among many other examples, as a testimony to its success and resilience.

FLOSS projects are often perceived to be of high quality [11], due to the adoption of an open development model, which promotes peer review, as suggested by several authors. In any case, companies are aware that integrating FLOSS into commercial products, made available by liberal OSS licenses, reduces development costs while offering high-quality, extensively tested components. Furthermore, Governments are aware of growing dependence on proprietary formats and software in their administration, and regard FLOSS as a warranty of technological independence.

However, state-of-the-art FLOSS, by the very nature of its open, unconventional, distributed development model, makes software quality assessment, let alone full certification, particularly hard to achieve and raises important challenges from both the technical/methodological and the managerial points of view. As a result, integration of FLOSS within industrial-strength applications, with stringent security requirements, is perceived as risky.

To address this issue one needs to have a precise idea of the dimension of the FLOSS reality and of the practices and tools oriented to quality analysis. According to Sands, GitHub has 4.751,000 repositories, SourceForge 324,000 projects and Ohloh 550,000 projects [12]. In other words, FLOSS is a huge phenomenon. However, communities are dynamic, often present irregular evolution and a variety of mostly ephemeral interactions. As depicted in Table I, which gathers data from Ohloh the open source network 0 until the end of March 2012, only a small fraction of all projects ever started gain long-term traction.

TABLE I: Ohloh Activity

Ohloh activity	
550,000+	Projects.
271,372	With a code analysis.
96,824	With a commit in the past 2 years
46,883	With a commit in the past year.
29,303	With a commit in the past 6 months.
21,251	With a commit in the past 3 months.
12,870	With a commit in the past 1 month.
5,629	With a commit in the past week.
1,224	With a commit in the past day.

Concerning quality assurance, there are many portals that can analyze FLOSS projects, such as, Squale, QSOS, and the Alitheia Core. These are actually FLOSS projects themselves, usually targeting a specific language or programming environment.

The CROSS portal [13], which is instrumental for this paper, is a web-based, open platform to analyze and certify FLOSS projects. Unlike other tools, it was not designed to a meet a specific target and is flexible enough to be tailored when particular needs of a user differ from the built-in analysis repertoire offered. The portal acts as a repository of tools, freely available, whose combined application produces reports upon which some form of certification of FLOSS products can be based. Different "test-tools" are plugged-in and made available as "code analyzers" to support FLOSS development. In brief, the portal is able to 1) register open source software projects 2) offer a number of analyzers for source code or documentation and guide their application on these projects, and 3) render analysis results in suitable, flexible formats to both FLOSS developers and users. In this way, the CROSS portal emphasizes a recovery – validation cycle, through a light analysis infrastructure which implements a number of techniques for code analysis and reverse engineering [14].

IV. The Role of CROSS

The CROSS portal has an open architecture and is, itself, managed in a collaborative way, right in the spirit of FLOSS communities. Our working hypothesis is that, when used from inside these communities, it may add to the informal learning networks such communities implicitly are, an analysis dimension to provide enhanced feedback to developers and enrich the community culture towards certification issues.

The problem domain is depicted through four elements representing different dimensions of the learning network that are of interest to this work: 1) the FLOSS community, 2) the CROSS portal, 3) FLOSS Projects, and 4) Formal Methods. Each of them and their relationships with the learning network are depicted in Fig. 1.

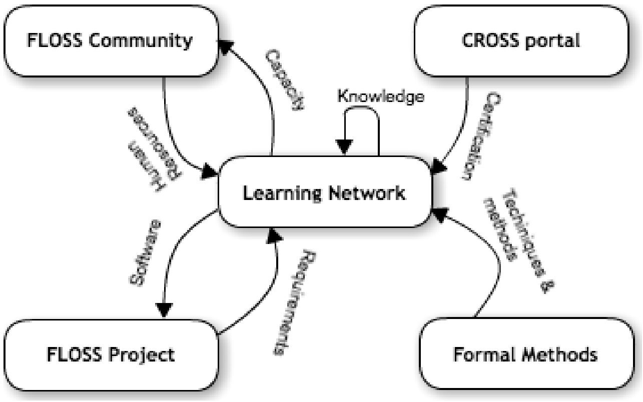

Fig. 1. The problem domain.

In this Fig. 1 Human Resources and Capacity represents the link between the Learning Network and the FLOSS community. The Human Resources provide knowledge to the Learning Network. Knowledge is shared and allows for building capacity in the community. FLOSS projects are developed by FLOSS communities. In order to develop a FLOSS project, it is necessary to share within the Learning Network its requirements, which usually evolve with the project. The final output is the software produced. During this phase of the process, new knowledge is developed and shared within the Learning Network. The use of Formal Methods code analysis techniques is motivated by the expectation that, as in other engineering disciplines, performing appropriate rigorous analysis contributes to the reliability and robustness of a design. During the application of formal techniques and methods knowledge is developed and shared. Finally, the CROSS portal supports the certification of software, such as producing new knowledge within the Learning Network, and builds higher user confidence in the community performance.

The introduction of an analysis and certification web-based platform which can be taken as a "working partner" in the dynamics of FLOSS communities enhances their patrimony as learning networks without disturbing the usual development cycles. On the other hand, from our perspective, it may be an effective way to introduce certification mechanisms and rigorous techniques in FLOSS development, meeting therefore the crucial demand from industry identified in Section II.

The nature of FLOSS entails the need for integration of techniques spanning the "micro" to the "macro" levels (e.g., from slicing to architectural recovery) and with different levels of formality (e.g. from statistical analysis based on code metrics to the identification and verification of hidden invariants). This is exactly what the CROSS portal aims at.

Rather than incorporating formal design and verification methodologies within the FLOSS development process, our emphasis is placed on reverse engineering techniques. Such techniques are fundamental to understand, document and validate FLOSS code. Moreover, through appropriate feedback loops they can contribute to the development of local, community-based learning networks.

To summarize, the combined use of certification techniques, metrics and tools within FLOSS communities has potential both to improve quality certification in open software and to enrich the community-shared culture. Moreover, to put this programme into practice, we foresee the following challenges: 1) introduce in FLOSS communities and development processes new techniques for analysis and the associated feedback loops; 2) support this process through suitable technology (namely, the CROSS portal) and popularize its use; 3) building new discovering and learning techniques inside FLOSS communities to assess experimental studies through qualitative methods based on case-studies and life-stories; and 4) generalizing the lessons learned within the FLOSS domain to other peer-production communities. The concept of FLOSS community can actually be generalized to the concept of open community by applying the idea of open source to a variety of collaborative efforts other than software production. What distinguishes an open community from a closed one is that anyone may join and contribute. Moreover, the direction and goals are determined collaboratively by all members of the community.

V. Conclusions

In this paper we claim that FLOSS development communities can be thought of as learning networks. Moreover, by integrating the CROSS certification portal, and the feedback loops it entails, such networks can build capacities for quality assessment and enforcement, with potential for a long-term impact on the integration of trustworthy, FLOSS components, in large, complex systems.

The experiment proposed here, is educationally relevant in the sense that it provides the community with a new kind of knowledge and savoir-faire, related to code analysis and certification, which adds to the whole educational potential of FLOSS projects. In broad terms, participation in FLOSS communities and projects can be regarded as a non-formal educational opportunity. Recent research projects, as documented by Cerone and Sowe [10], are exploring its relevance as a complement for formal education in Software Engineering. In these contexts, the introduction of further technology-enhanced practices, as discussed here, also opens new paths for the future. Of course, new practices need to be informed by research in learning theory and suitable pedagogical frameworks and by exploring FLOSS peer-production models as both a cultural phenomenon and an opportunity for innovative educational practices in the years to come.

References

[1] S. Scott and A. Palincsar. (23 November 2012). Sociocultural theory. [Online]. Available: http://www.education.com/reference/article/sociocultural-theory/

[2] 3 Reasons Why E-Learning Is Bigger and Better than Ever. (23 November 2012). [Online]. Available: http://www.elearners.com/online-education-resources/online-learning/3-reasons-why-e-learning-is-bigger-and-better-than-ever/

[3] J. S. Brown and R. P. Adler, "Minds on fire: open education, the long tail, and learning 2.0," *Educause Review Magazine*, vol. 43, no. 1, pp. 16-32, 2008.

[4] M. S. Reed, A. C. Evely, G. Cundill, I. Fazey, J. Glass, A. Laing, J. Newig, B. Parrish, C. Prell, C. Raymond, and L. C. Stringer. (2010). What is social learning? *Ecology and Society*. [Online]. 15(4) Available: http://www.ecologyandsociety.org/vol15/iss4/resp1/

[5] Towards new learning networks. (21 November 2012). [Online]. Available: http://www2.futurelab.org.uk/resources/publications-reports-articles/opening-education-reports/Opening-Education-Report121

[6] Ohloh. (22 November 2012). The open source network. [Online]. Available: http://www.ohloh.net/

[7] C. R. Hollenbeck, C. Peters, and G. M. Zinkhan. "Gift giving: a community paradigm," *Psychology and Marketing*, vol. 23, no. 7, pp. 573-595, 2006.

[8] L. Dirckinck-Holmfeld, C. Jones, and B. Lindström. "Analysing Networked Learning Practices in Higher Education and Continuing Professional Development," *Rotterdam: Sense Publishers, BV,* 2009, pp. 5-6.

[9] University of the first age: celebrating learning. (23 November 2012). [Online]. Available: http://www.ufa.org.uk

[10] A. Cerone and S. K. Sulayman, "Using Free/Libre Open Source Software Projects as E-learning Tools," in *Pre-proceedings of the 4th International Workshop on Foundations and Techniques for Open Source Software Certification (OpenCert 2010)*, pp. 154, 2010.

[11] M. Michlmayr, F. Hunt, and D. Probert, "Quality practices and problems in free software projects," presented at the First International Conference on Open Source Systems, Genova, Italy, July 11-15, 2005.

[12] R. Sands, "Open Source By The Numbers," *Black Duck Software*, 2012.

[13] P. Martins, J. P. Fernandes, and J. Saraiva, "A web portal for the Certification of Open Source Software," presented at OpenCert 2012, Thessaloniki, Greece, October 2, 2012.

[14] Reverse Engineering. (21 November 2012). [Online]. Available: http://en.wikipedia.org/wiki/Reverse_engineering

Sara Fernandes is a PhD student at the International Institute of Software Technology of the United Nations University in Macau and the MAP-I Doctoral Programme at Minho, Aveiro and Porto Universities, Portugal. She is doing research on Peer-Production and Technology-Enhanced Collaborative Teaching and Learning.

Luis S. Barbosa is an associate professor at the Informatics Department of Universidade do Minho, Portugal, and a researcher at INESC TEC in high-assurance software and formal methods. He coordinates the joint Doctoral Programme in Computer Science of the Universities of Minho, Aveiro and Porto.

Antonio Cerone is Research Fellow at the International Institute for Software Technology of the United Nations University. His research focuses on education and e-learning as well as on formal methods and their applications. He is the leader of the UNU-IIST Programme on ``Strategies and Technologies for Education" and coordinates the UNU-IIST Postgraduate Programme.