Facilitating Social Network Studies of FLOSS using the OSSNetwork Environment

Marco A. Balieiro, Samuel F. de Sousa Júnior, and Cleidson R. B. de Souza
Faculdade de Computação – Universidade Federal do Pará
66075-110 – Belém – PA – Brazil,
{ma.balieiro, sfelixjr}@gmail.com, cdesouza@ufpa.br

Abstract. Open source projects are typical examples of successful distributed software development projects. Understanding how coordination in these projects takes place can provide important lessons to Software Engineering researchers and practitioners. This understanding has been achieved using different research methods, including, surveys, case studies and social network analysis. However, to conduct these studies each researcher needs to build his own infra-structure from the scratch, a time consuming and error-prone task. This paper aims to alleviate this problem by describing an environment, the OSSNetwork, which allows the automatic data collection of open source repositories. Data collected by the OSSNetwork is aimed to support the construction, visualization, and analysis of social networks. This environment is extensible, therefore facilitating empirical studies of open source projects.

1 Introduction

The Free/Libre Open Source Software (FLOSS) movement has become an economically viable and financially satisfactory alternative to proprietary software due to its reduced cost, good performance in critical operations and data manipulation, and improved security [1]. Supporters of FLOSS argue that the availability of the source code influences positively its quality, since any developer can review the code and improve it [2]. Today, the Internet infrastructure has great part of its critical elements based on FLOSS [3].

Due to these factors, researchers and practitioners of several areas (software engineering, economy, sociology, etc) are interested in understanding FLOSS projects from different viewpoints, including developers' motivation [4], the software process adopted by these communities [5], quality assurance [6], just to name a few. To address these different goals, several research methods have been used, from ethnography [7], to case studies [8], social network analysis [9], and even traditional statistical methods [10]. In particular, the use of the Social Network Analysis (SNA) allows the study of relationships among developers in these communities. For instance, Lopez-Fernandez, *et al*. [11] used social networks to understand the social relationships based on data available in CVS repositories, whereas Crowston and Howison [12] studied the networks of people who got

involved in bug fixing activities, and de Souza, *et al*. [13] analyzed developers' social network extracted from source-code dependency relationships.

All these approaches helped to understand important aspects of FLOSS projects. However, a limitation of these approaches is that they were conducted independently: each researcher had to build his own tools to extract, manipulate and analyze the data. To minimize this effort, Howison, *et al*. [14] developed the FLOSSmole, a tool for collect, store and distribute FLOSS data and analysis. While FLOSSMole alleviates part of this problem, researchers still face difficulties to collect data from FLOSS repositories to perform social network analysis such as: collection and storage the data which are in servers that researchers do not have direct access, differences between data models from different repositories, handling large amounts of information, treatment of specific information that can generate ambiguities (such as the problem of aliases [15]), implementation of algorithms on the data collected or even on the social networks generated, and visualization and manipulation of these networks.

The work described in this paper extends the FLOSSmole tool with an environment, called OSSNetwork, which allows the study of FLOSS communities using social network analysis. This environment is extensible so that new algorithms, visualization, and functionalities can be added. Social networks are generated with information extracted from mailing lists, forums, issue-tracking tools and chat logs. All this information is stored in a local database for future analysis. We argue that the problems faced by SNA researchers described in the previous paragraph are minimized by using the OSSNetwork environment.

The rest of this paper is organized as follows. In the next Section we will briefly present our motivation to this work. Next, a very short review of social network definition will be presented. The following section describes the OSSNetwork environment, and is followed by a case study and a discussion of the obtained results. Finally, we present our conclusions and future work.

2 Motivation

Howison, *et al*. [14] describes a research process to be used when studying FLOSS projects. This process is presented in Figure 1. According to this process, one of the most difficult tasks is the selection of projects to analyze. According to Howinson, two mechanisms can be used: *census*, where all existing projects of a particular phenomenon to be examined must be used, which is very difficult because no one knows how many projects exists; and *sampling*, where a small number of projects that represents a particular phenomenon is selected randomly. This is not an easy task, because researchers actually end up limiting their studies to a single repository. More importantly, sampling open source projects is methodologically difficult because everything FLOSS research has shown so far points to massively skewed distributions across almost all points of research interest [16][17] and even

randomly selecting the projects, still, in general, does not produce a representative sample [14].

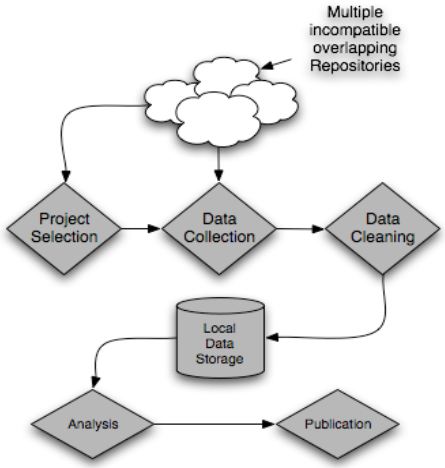

Figure 1 – Research process in FLOSS projects

Project selection becomes even more complicated because public repositories contain a large number of inactive, relocated or disabled projects. Our approach to handle this problem, detailed later, is to expand the number of supported repositories. After selecting the projects, two processes can be used for data collection: getting the databases *dumps* of the repositories or conduct a *parsing* of the repositories' web pages. Although gaining access to the databases is clearly preferable, not all repositories make their dumps available. Therefore, parsing is the data collection method that is made available by OSSNetwork, because it only requires the availability of Internet access.

3 A brief Social Network overview

A social network is broadly defined as a set of relationships [18]. A social network has a set of objects (nodes, in mathematical terms) and a description of the relationships between these objects. For example, it can be said that nodes are the people of a house and the relation that establish connection between these people is "people who use the same room".

A social network can be characterized according to structural and topological properties [19]. These properties are derived from the graph theory. The structural properties are: node degree, weighted degree of a node, distance centrality of the node, proximity degree, betweenness centrality, and others. Topological properties include: density of the network, distribution degree, network diameter, and finally,

cluster degree, that it is a set of connected nodes through some way, them is considered as representative of communities.

Social networks can be classified according to two types: *1-mode networks*, which represent the relationship between social entities of the same type, for example, who is friend of who, who depends on whom; and *2-mode networks*, which represents relationships between different social entities, for example, the people who had been to a meeting, the developers that had corrected a given bug. It is important to underline here that from a 2-mode network, it is possible to easily generate a 1-mode network [20].

4 The OSSNetwork Environment

The OSSNetwork environment allows one to: (*i*) retrieve information from FLOSS repositories, (*ii*) store this information in a database, (*iii*) generate different social networks from this information, and, finally, (*iv*) analyze these networks using tools to manipulate, edit, and execute algorithms. This can be seen in the Figure 2.

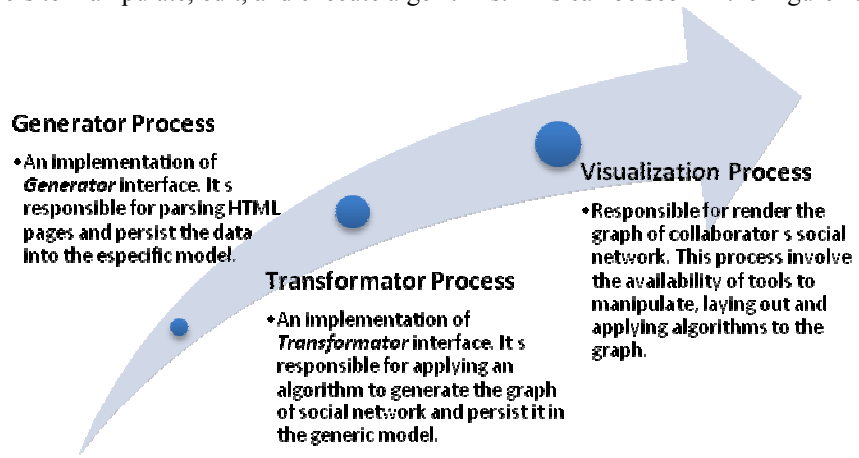

Figure 2 – The OSSNetwork approach

The OSSNetwork is implemented in Java and uses matrices to generate the social networks from the data extracted from projects repositories. Data is extracted through parsing HTML information about forums, mailing lists, bug tracking and chat.

4.1 An Example: the Apache Jackrabbit project

Figure 3 below illustrates the social network generated with the OSSNetwork environment from the messages exchanged in the mailing list of Apache Jackrabbit

project. In this figure, it is possible to observe some features already implemented, such as: handling of vertices and edges, use of geometric shapes and sizes according to some metrics, annotations, highlight of neighbor vertices, etc.

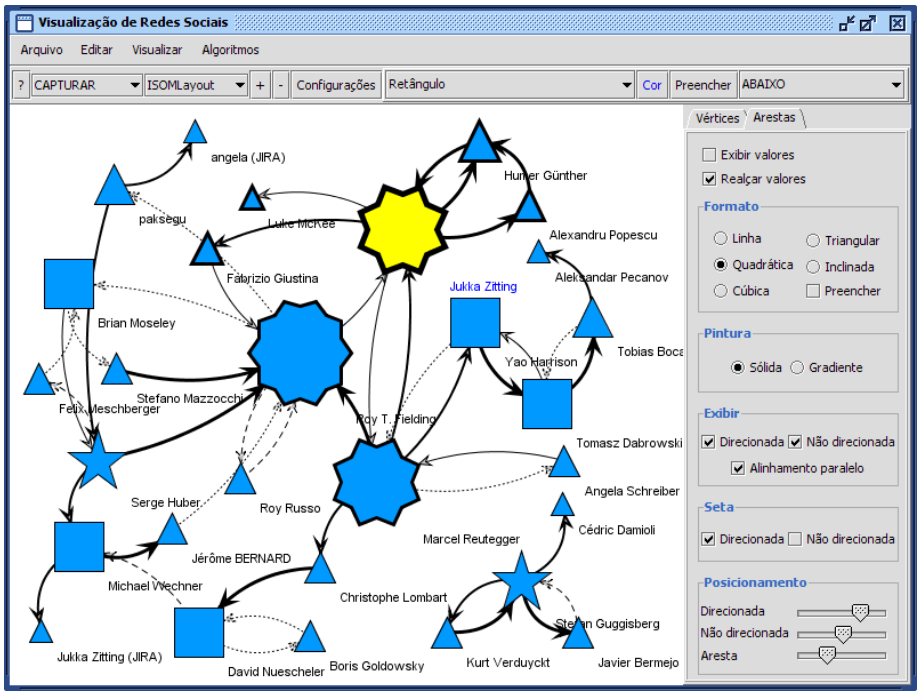

Figure 3 – The Apache Jackrabbit project in the OSSNetwork environment

4.2 Extracting Information from FLOSS Repositories

The OSSNetwork aims to minimize the effort of researchers interested in the social network analysis of FLOSS communities, which requires the extraction of information from these communities. Currently, the OSSNetwork has support for several repositories including: SourceForge, RubyForge Apache, and Microsoft Codeplex.

As mentioned earlier, the environment uses the FLOSSmole framework [14] to parse HTML web pages from the supported repositories. In addition, we implemented new parsing abilities to extract information to be used for social network analysis, for instance, information from the mailing lists which includes who sent a message, who replied to that message, etc. All these steps can be extended providing a new implementation of the *Generator* interface.

4.3 Social Network Generation

Social networks are generated from information extracted from mailing lists, forums and the discussion messages associated with bugs. The mechanism used to generate these networks uses two matrix operations: multiplication and transposition. These operations are required to obtain a 1-mode network from an existing 2-mode network [20]. All generated data related to social networks are stored in a new generic model.

Addition new algorithms to generate different social networks is an easy task. In order to do that, one needs to implement the *Transformator* interface. The *transform* method receives the list of discussion and must execute a calculation over the data and return a graph represented by the generic model of social network that is based on the JUNG framework [21]. Social networks generated by the environment can be exported to files in XML, CSV and DL formats to be used in other social network analysis tools like UCINet [22].

5 Conclusions and Future Work

Empirical research on FLOSS has aroused increasing interest from researchers. Nevertheless, tools that assist in this type of research are still scarce in view of the difficulties inherent in the research process. The OSSNetwork environment described in this paper helps to reduce these difficulties by providing an environment with an integrated set of tools and functionalities to facilitate data extraction, manipulation, and analysis. These functionalities can be extended by new implementations of some interfaces and adding new algorithms to the set of tools currently supported.

We expect that our environment will facilitate the adoption of new approaches using social network analysis. In particular, we are interested in multi-dimensional analysis of social networks extracted from FLOSS communities, that is, analysis where it is possible to take into account, at the same time, different social networks created from different data, such as a public forum, chat rooms, bug tracking systems, mailing lists, etc. We argue that a multidimensional analysis will allows us to study the relationship among different social networks. For instance, the different roles that a same developer has on different aspects (code, bug fixing, user support, etc) of a FLOSS project.

Acknowledgments

This research was supported by the Brazilian Government under grant CNPq 479206/2006-6, by the Universidade Federal do Pará, and by a Microsoft grant.

References

[1]. **Hoepman, J. and Jacobs, B.** *Increased Security Through Open Source.* Communications of the ACM, pp. 79-83, 2007.

[2]. **Raymond, E. S.** *The Cathedral and the Bazaar.* s.l. : First Monday, 1998.

[3]. **Madey, G., Freeh, V. and Tynan, R.** Modeling the F/OSS Community: A Quantitative Investigation. *Free/Open Source Software Development.* s.l. : Idea Publishing, 2004.

[4]. **Rullani, F.** *Dragging Developers Towards The Core. How The Free/Libre/Open Source Software Community Enhances Developer's Contribution.* Pisa, Italy, 2006.

[5]. **Jensen, C. and Scacchi, W.** *A Reference Model for Discovering Open Source Software Processes.* Limerick, IR. Third IFIP International Conference on Open Source Systems, 2007.

[6]. **Halloran, T. and Scherlis, W.** *High Quality and Open Source Software Practices.* Orlando, FL : s.n., Workshop on Open Source Software Engineering, 2002.

[7]. **Ducheneaut, N.** *The Reproduction of Open Source Software Programming Communities.* Berkeley, CA : UC Berkeley, School of Information Management and Systems, 2002.

[8]. **Jensen, C. and Scacchi, W.** Role Migration and Advancement Processes in OSSD Projects: A Comparative Case Study. *International Conference Software Engineering.* 2007.

[9]. **Gao, Y. and Madey, G.** *Network Analysis of the SourceForge.net Community.* Limerick, Ireland : s.n., International Conference on Open Source Systems, 2007.

[10]. **Mockus, A., Fielding, R. and Herbsleb, J.** *A Case Study of Open Source Software Development: The Apache Server.* Limerick, IR, International Conference on Software Engineering, 2000.

[11]. **Lopez-Fernandez, L., Robles, G. and Gonzalez-Barahona, J.** *Applying Social Network Analysis to the Information in CVS Repositories.* Juan Carlos, Spain : Universidad Rey Juan Carlos, 2004.

[12]. **Crowston, K. and Howison, J.** *The Social Structure of Open Source Software Development Teams.* Seattle, WA : s.n., International Conference on Information Systems, 2003.

[13]. **De Souza, C., Froehlich, J. and Dourish, P.** *Seeking the Source: Software Source Code as a Social and Technical Artifact.* Sanibel Island, FL : s.n., ACM Conference on Group Work, 2005.

[14]. **Howison, J., Conklin, M. and Crowston, K.** FLOSSmole: A Collaborative Repository for FLOSS Research Data and Analyses. *Journal of Information Technology & Web Engineering.* 2006.

[15]. **Gertz, M.** *Mining Email Social Networks in Postgres.* Shanghai, China : s.n., International Workshop on Mining Software Repositories, 2006.

[16]. **Conklin, M.** *Do the Rich Get Richer? The Impact of Power Laws on Open Source Development Projects.* Portland, Oregon : s.n., Open Source Conference (OSCON), 2004.

[17]. **Xu, J., et al.** *A Topological Analysis Of The Open Source Software Development Community.* Big Island, Hawaii : IEEE Computer Society, HICSS, 2005.

[18]. **Kadushin, C.** *Introduction to Social Network.* 2004.

[19]. **Hanneman, R. and Riddle, M.** *Introduction to Social Network Methods.* Riverside, CA : University of California, 2005.

[20]. **Wasserman, S. and Faust, K.** *Social Network Analysis: Methods and Applications.* Cambridge, UK and New York : Cambridge University Press, 1997.

[21]. **Fisher, D., et al.** Analysis and Visualization of Network Data Using JUNG. *Java Universal Nerwork/Graph Framework.* [Online] http://jung.sourceforge.net/doc/JUNG_journal.pdf.

[22]. **Borgatti, S., Everett, M. and Freeman, L.** UCINET 6 Social Network Analysis Software. *Analytic Technologies -- Social Network Analysis & Cultural Domain Analysis.* 2006.