iadis international conference

IADIS MULTI CONFERENCE ON COMPUTER SCIENCE AND INFORMATION SYSTEMS

MCCSIS 2009

21 - 23 June
Algarve, Portugal

Proceedings of ICT, SOCIETY AND HUMAN BEINGS 2009 and WEB BASED COMMUNITIES 2009

Edited by:
Gunilla Bradley
Piet Kommers

iadis
international association for development of the information society

IADIS INTERNATIONAL CONFERENCE

ICT, SOCIETY AND HUMAN BEINGS 2009

and

IADIS INTERNATIONAL CONFERENCE

WEB BASED COMMUNITIES 2009

part of the

IADIS MULTI CONFERENCE ON COMPUTER SCIENCE AND

INFORMATION SYSTEMS 2009

PROCEEDINGS OF THE

SECTION I

IADIS INTERNATIONAL CONFERENCE ICT, SOCIETY AND HUMAN BEINGS 2009

SECTION II

IADIS INTERNATIONAL CONFERENCE WEB BASED COMMUNITIES 2009

part of the

IADIS MULTI CONFERENCE ON COMPUTER SCIENCE AND INFORMATION SYSTEMS 2009

Algarve, Portugal

JUNE 21 - 23, 2009

Organised by
IADIS
International Association for Development of the Information Society

Volume Editors:
Gunilla Bradley and Piet Kommers

Computer Science and Information Systems Series Editors:
Piet Kommers, Pedro Isaías and Nian-Shing Chen

Associate Editors: Luís Rodrigues and Patrícia Barbosa

ISBN: 978-972-8924-82-9

SUPPORTED BY

FCT Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA CIÊNCIA E DO ENSINO SUPERIOR Portugal

SECTION II

IADIS INTERNATIONAL CONFERENCE

WEB BASED COMMUNITIES 2009

part of the

IADIS MULTI CONFERENCE ON COMPUTER SCIENCE AND INFORMATION SYSTEMS 2009

SHORT PAPERS

REFLECTION PAPERS

ISBN: 978-972-8924-82-9 © 2009 IADIS

ANALYSIS OF ONLINE PARTICIPATION IN OPEN SOURCE COMMUNITIES

M. R. Martínez-Torres
E. U. Estudios Empresariales, University of Seville
Avda. San Francisco Javier, s/n 41018 Sevilla (Spain)

S. L. Toral, F. Barrero
*E. S. Ingenieros, University of Seville **
Avda. Camino de los Descubrimientos s/n, 41092, Seville, Spain

ABSTRACT

The success of an Open Source Software (OSS) project is closely linked to the successful organization and development of the virtual community of support. In particular, participation is the most important mechanism in which the development of the project is supported. The main objective of this paper is to analyze the online participation in virtual communities using Social Network Analysis (SNA) techniques. Several Open Source communities related to Linux ports to embedded processors are studied, obtaining a set of indicators from their social network representation. The statistical analysis of these indicators allows the extraction of several conclusions about the online participation in this kind of virtual communities.

KEYWORDS

Open Source, Social Network analysis, virtual communities, factor analysis.

1. INTRODUCTION

Open source software (OSS) is software whose source code is available to users and can be distributed with few limitations on possible modifications and distribution by third parties (Open Source initiative, 2002). In particular, OSS projects are developed and released under some sort of ‘open source’ license that allows inspection and reuse of the software’s source code (Crowston & Scozzi, 2002). Each OSS project is supported by a few, dozens or even hundreds of geographically distributed developers, organized as an Internet-based community, who voluntarily collaborate to develop the underlying software. The success of OSS projects has been attributed to their speed of development and the reliability, portability, and scalability of the resulting software. These claimed advantages of OSS development are due to the fact that the source code is open to the Internet community. Since everybody can access and review anybody else’s work, developers can learn from each other and improve their overall software development skill (Lussier, 2004). Also, instead of using huge financial resources to put the software through extensive testing and Quality Assurance, like a proprietary vendor will do, the Open Source projects have the community as a resource (Lakhani and Hippel, 2003). Finally, several studies claim that OSS is developed faster, cheaper, and the resulting systems are more reliable than proprietary software (Mockus et al., 2002).

This paper will be focused on successful OSS development, but instead of analyzing the development in terms of the source code produced, we will focus on the social relationships among virtual community members. Virtual communities have become an important new organizational form and yet relatively little is known about the conditions which lead to their success. It is usually assumed that OSS communities are organized in a certain structure in which their members perform different roles according to their degree of involvement. This paper goes beyond this idea defining to what extent the structure and the different members’ profiles have an incidence on the successful development of the OSS community. Social network analysis (SNA) techniques will be used for this purpose. SNA models the community as a group of participants considering the links among them. From this model, several indicators measuring some features of the community can be derived. A case study based on the Linux ports to different processors will be used to measure the proposed indicators and to identify the main dimensions involved in the development of the community. The obtained results could be applied to design or to improve OSS communities.

2. VIRTUAL COMMUNITIES IN OSS PROJECTS

Virtual communities can be studied from the perspective of Communities of Practice (CoP) developed by Lave & Wenger (1991). This concept refers to the process of social learning that occurs when people who have a common interest in some subject or problem collaborate over an extended period to share ideas, find solutions, and build innovations. The basic assumption underlying the theory of CoPs is that engagement in social practice is the fundamental process by which we learn (Wenger, 1998). Learning in community should be viewed as an integral constituent of participation in the CoP, and as a process of constructing knowledge through social interaction with other members of the community, of changing relationships with other members of the community, and of transforming roles and establishing identities from a journeyman to a master in the community (Rohde et al., 2007). When interactions take place using electronic media, these communities are often referred as virtual communities or networks of practice (Johnson, 2001).

OSS communities are typically initiated by an individual (or group of individuals) who provides systems and development components, or their access, as well as communication infrastructure. Participants are usually volunteers and contributors are not normally motivated by traditional economic incentives, but rather by instrumental factors associated with fulfilling a need, and by intrinsic factors such as enhanced reputation, expertise development (learning), self-fulfillment, as well as basic fun and enjoyment (Lerner and Tirole, 2002). The individuals that participate in open source software projects are often described as comprising a community. These communities have been described as having an onion-like structure, with a central core of highly active individuals, surrounded by other layers of progressively less active individuals. It has been demonstrated that much of the OSS development is realized by a small percentage of individuals despite the fact that there are tens of thousands of available developers. Such concentration is called "participation inequality" (Kuk, 2006), and it can be explained by the different user profiles of open source communities. Consequently, the structure of OSS communities is not completely flat as it was claimed by the bazaar model of full participation (Raymond, 1998). This argument is based on the logic that a highly participative community may lead to richer discussion, better flow of ideas, efficient code development, faster bug finding and fixing, and, hence, faster and efficient project growth (Weber, 2004). However, contrary to the bazaar view of OSS projects, many empirical studies have found that only a small number of developers contribute a large percentage of code and discussion in OSS projects. Participation inequality allows the categorization of OSS community members in three groups (Mockus et al., 2002):

- Core members. They are responsible for guiding and coordinating the development of an OSS project. They are usually involved with the project during a long period of time and have made significant contributions to the development and evolution of the system.
- Active developers. They regularly make contributions to the project.
- Peripheral developers. They occasionally contribute with new features to the existing system. This contribution is irregular, and the period of involvement is short and sporadic. Free riders (people who just are seeking answers without making any contributions) are also included in this group.

The roles and their associated influences in OSS communities can be realized only through contributions to the community. In this paper, we will analyze mailing lists because they allow the collective reflection and community discussions, and activities are not just confined to software development or coding alone (Sowe et al., 2006). Interactions are usually structured in threads of discussion, which facilitates their analysis.

The simplest way to classify threads is using their length, i.e. the total number of posts they contain. Posts per thread turn out to be a reliable metric to determine the degree of "conversational concentration" of an author in a given group (Bonacci, 2004). Nevertheless, this kind of data does not provide any information about the social structure of the community, or about the relationships among authors. In this paper, social networks will be extracted from threads of discussion, and SNA techniques will be applied to provide new insights in the community organization (Cho et al., 2005). A social network can be represented as a graph $G = (V,E)$ where V denotes a finite set of vertices and E denotes a finite set of edges such that $E \subseteq V \times V$.

In the context of threads of discussion, V is given by all the authors posting messages and E is given by the successive answers among authors inside a thread, which is the basic unit considered (Jones et al., 2004). The use of discussion threads as the basic unit of analysis is highly valid, considering that the epistemic interactions in support of OSS development often take place in discussion threads where individual postings provide the context to encourage participation (Kuk, 2006). In contrast to a reply to a single message, it is more cognitively complex to reply to a threaded discussion, because the ebb and flow of earlier postings must be taken into account to develop a coherent answer (Knock 2001). That is the reason why an author posting to a thread will be tied to all the authors who have previously posted to the same thread when constructing the social network. The resulting graph will exhibit the following features:

- It will be a directed graph. The direction of the arc is given by the flow of information between two authors. That means that a sender (the tail of the arc) is answering a receiver (the head of the arc) inside a thread of discussion.
- It will be a valued graph. An author is able to participate several times inside a thread or can answer to the same authors in different threads.

Networks can be partitioned using some discrete characteristics of vertices. For instance, several classes of vertices can be obtained using the value of arcs. In the case of OSS projects, these kinds of partitions should highlight the core/periphery (C/P) structure of the community. A C/P structure divides vertices in two distinct subgroups: vertices in the core, densely connected with each other, and vertices on the periphery, not connected with each other, only nodes in the core. In network analysis, density is a measure of the cohesion of the network. More ties between people yield a tighter structure, which is, presumably, more cohesive. Density can be defined as the number of lines in a simple network, expressed as a proportion of the maximum possible number of lines. Consequently, maximum density is found in a network where all pairs of vertices are linked by two arcs, one in each direction. Sometimes, network density is not very useful because it depends on the size of the network. In this case, it is better to look at the number of ties in which each vertex is involved. This is called the degree of a vertex. As we are involved with a directed network, we will actually use the concept of out-degree of a vertex, that is, the number of arcs it sends. Therefore, the average out-degree of all vertices could be used to measure the structural cohesion of a network.

3. CASE STUDY

The case study is based on Linux ports to embedded processors. Linux is a PC-based operating system that has been developed as Open Source Software along the structure of the UNIX operating system, and it is one of the most prominent examples of OSS projects. Although Linux started as a hobby in 1991, it represents today a serious threat to Microsoft Windows's market dominance in operating systems (Cusumano and Selby, 1997). Nevertheless, the proposed case study will be focused on Linux ports to other processor architectures not intended for desktop or personal computer market. There are several reasons for this choice. First, Linux is firmly in first place as the operating system of choice for smart gadgets and embedded systems. Second, in contrast to other typical open source projects or even desktop Linux project, most contributions in this field do not come from volunteers or hobbyists, but from commercial firms, many of which are dedicated embedded Linux firms. Third, there are a lot of communities supporting each one of these Linux ports, and this is an excellent opportunity for analyzing a big group of more or less "homogeneous" communities. Up to eleven virtual communities have been considered:

- The ARM Linux Project (ARM). ARM Linux is a port of the successful Linux Kernel to ARM processor based machines.
- Debian port to ARM (D-ARM). ARM port for Debian GNU/Linux. Debian fully supports a port to little-endian ARM.
- Linux PPC port (PPC). PowerPC Linux is the Linux kernel running on a PowerPC processor.
- Debian port to PowerPC (D-PPC). PowerPC port of Debian GNU/Linux. The PowerPC architecture allows both 64-bit and 32-bit implementations.
- Debian port to m68k (D-68k). Motorola 68k port of Debian GNU/Linux. Debian currently runs on the 68020, 68030, 68040 and 68060 processors.
- Debian port to Alpha (D-Alpha). The purpose of this project is to assist developers and others interested with the ongoing project to port the Debian distribution of Linux to the Alpha family of processors.

- Debian port to MIPS (D-MIPS). MIPS port of Debian GNU/Linux, able to run at both endiannesses.
- Debian port to BSD (D-BSD). This is a port of the Debian operating system, complete with apt, dpkg, and GNU userland, to the NetBSD kernel.
- Debian port to HPPA (D-HPPA). This is a port to Hewlett-Packard's PA-RISC architecture.
- Debian port to Hurd (D-HURD). The GNU Hurd is a totally new operating system being put together by the GNU group.
- Debian port to SPARC (D-SPARC). This port runs on the Sun SPARCstation series of workstations, as well as some of their successors in the sun4 architectures.

Nine of them are Debian Linux ports to different processor architectures. The Debian Project is an association of individuals who have made common cause to create a free operating system called Debian GNU/Linux. The other two virtual communities are specific Linux ports to ARM and PowerPC processors.

Each community will be analyzed during the period 2001-2007, which is the common period in which all the considered communities have been active. For each year and community, a social network based on interactions among participants has been extracted. As a result, a total of 77 social networks have been analyzed. The out-degree of each vertex will be used to distinguish among the different community members profiles. In particular, those members with an out-degree higher than the average out-degree of the social network will be considered as active contributors and those members with an out-degree higher than this average value plus the standard deviation will be considered as core members. Notice that these threshold values are chosen arbitrarily, but the important question for the subsequent analysis is to define a way of distinguishing the different members' profiles independently of the size of the community. Using these general guidelines, the following variables can be extracted:

- Community out-degree: out-degree of a social network represents the degree of interactions in threads of discussion. Consequently, average and standard deviation out-degree values (V1 and V2) will be obtained to be used as a threshold to distinguish among peripheral, active and core developers.
- Active developers: the absolute value of active developers (V3) and their percentage respect to the whole community (V4) will be evaluated to consider the specific weight of this group.
- Betweenness: it is a measure of centrality that rests on the idea that a person is more central if he or she is more important as an intermediary in the communication network (Nooy et al., 2005). The centrality of a person depends on the extent to which he or she is needed as a link in the chains of contacts that facilitate the spread of information within the network. The more a person is a go-between, the more central his or her position is in the network. If we consider that the shortest path between two vertices (geodesic) is the most likely channel for transporting information between actors, an actor who is situated on the geodesics between many pairs of vertices is very important to the flow of information within the network. The betweenness centrality of a vertex is the proportion of all geodesics between pairs of other vertices that include this vertex, and betweenness centralization of the network is the variation in the betweenness centrality of vertices divided by the maximum variation in betweenness centrality scores possible in a network of the same size (Nooy et al., 2005). Two values of betweeness will be considered: the betweeness of the whole network (V5) and the betweenness of the sub-network of active developers (V6).
- Core developers: the absolute value of core developers (V7) and their percentage respect to sub-network of active developers (V8) and the whole community (V9) will be evaluated to consider the specific weight of this group.
- Active and core developers out degree: the average out degree value of the sub networks of active developers (V10) and core developers (V11) are measures of participation inequality. The relative importance of the core will be measured evaluating the percentage of the out degree due to the core members of the community (V12), and their role as brokers (V13) or mediators among other core members.

A factor analysis will be applied to extract the main dimensions related to online participation in virtual communities. Factor analysis has been performed using the principal component method. The eigenvalues of the sample covariance matrix are shown in **Table 1**. In factor analysis it is usual to consider a number of factors able to account for more than 70% of the total sample variance. In our case, study, this value is achieved with three factors.

Table 1. Total variance explained

Factor	Eigen-values	% of Variance	Cumulative %
1	5,187	39,901	39,901
2	2,319	17,839	57,740
3	1,882	14,478	72,218
4	1,061	8,161	80,379
5	,831	6,395	86,774
6	,593	4,562	91,336
7	,383	2,947	94,283
8	,280	2,151	96,434
9	,209	1,604	98,038
10	,155	1,192	99,230
11	,051	,390	99,620
12	,033	,250	99,870
13	,017	,130	100,000

Table 2. Rotated Component matrix with Varimax rotation.

	F1	F2	F3
V1	,929	,226	,075
V2	,821	,143	,154
V3	,318	,699	,436
V4	-,251	,563	,513
V5	,283	,196	,721
V6	,014	-,244	,795
V7	,189	,923	-,054
V8	-,066	-,174	-,794
V9	,118	-,135	-,056
V10	,968	,104	-,001
V11	,963	-,009	,052
V12	-,653	-,138	-,436
V13	,365	,866	-,101

Using the associated eigenvectors, factor loadings can be estimated. Sometimes, it is difficult to perform the right interpretation of factors using the estimated loadings. Fortunately, factor loading can be rotated through the multiplication by an orthogonal matrix, preserving the essential properties of the loadings. Varimax method is an orthogonal rotation method that minimizes the number of variables that have high loadings on each factor. This method simplifies the interpretation of the factors. **Table 2** reports the rotated factor loadings with varimax rotation for each one of the economical areas analyzed.

To extract the meaning of each factor, we move horizontally through **Table 2**, from left to right, across the three estimated loadings of each variable, identifying the highest loading and the corresponding factor. To assess significance of factor loadings, a threshold value of 0,7 was considered. The association between variables and factors is highlighted in grey in **Table 2**. The resulting aggregation of variables leads to the following latent factors or dimensions:

The first factor is explained by the participation inequality typical of virtual communities. This factor exhibits a high value in variable V1 and V2, which corresponds to the average and standard deviation of the out-degree of the network. The high value of the standard deviation means that there is a great variability in participation among community members. The high value of V10 and V11 confirm that the group of active and core developers are responsible of the majority of contributions.

The second factor is related to the role of the core of the community. The core group is essential for the continuity of the community and they must play a brokerage role among contributors.

The third factor is related to the topology of the community. Centrality (V5 and V6) and the structure of the communities (V8) are included in this factor. The negative value associated to V8 means that the core group should be just a small fraction of active developers, to guarantee a good coordination of the community. Several implications can be derived from the obtained latent factors:

- The necessity of a participation inequality with a clear distinction between peripheral and active contributors. Open source communities are frequently visited by a lot of users just interested in asking for information, but with no intention of becoming an active contributor. Just a small fraction of visitors will become an active contributor as they learn through online participation. Learning does not appear as a result of being taught, but through direct engagement in the social, cultural, and technical practice of the community.
- The key role of the core group. The mission of the core group is not just participating but, above all, promoting the debate and participation and addressing the future development of the underlying project.
- Finally, a network structure is also necessary to achieve a good development of the community. The cohesion of the network is supporting the mechanism of participation, necessary for the project development, and participation promotes the success of the underlying project increasing the number of threads and contributions. On the other hand, cohesion also means a cohesive core group.

4. CONCLUSION

Communities are basically based on interactions among users, and participation is the basic mechanism promoting its development. This participation has been analyzed using social network analysis techniques. Several indicators related to features like cohesion, structure, centralization and user profiles have been obtained for a set of online communities related to Linux ports and then analyzed using factor analysis. The obtained results reveal three main characteristics of open source virtual communities, like participation inequality, the role of the core group of developers and the necessity of a certain centralized structure around a small number of core developers.

ACKNOWLEDGEMENT

This work has been supported by the Spanish Ministry of Education and Science (reference DPI2007-60128) and the Consejería de Innovación, Ciencia y Empresa (reference P07-TIC-02621)

REFERENCES

Bonacci, D., 2004. Towards quantitative tools for analysing qualitative properties of virtual communities. *Interdisciplinary Description of Complex Systems*, Vol. 2, no. 2, pp. 126-135.

Cho, H., Lee, J.-S., Stefanone, M. and Gay, G., 2005. Development of computer-supported collaborative social networks in a distributed learning community. Behaviour & Information Technology, Vol. 24, no. 6, pp. 435 – 447.

Crowston, K. and Scozzi, B., 2002. Open source software projects as virtual organisations: competency rallying for software development. *IEE Proceedings –Software*, Vol. 149, no. 1, pp. 3-17.

Cusumano, M. and Selby, R., 1997. How Microsoft Builds Software. *Communications of the ACM*, 40, no. 6, pp. 53-61.

Johnson, C.M., 2001. A survey of current research on online communities of practice. Internet and Higher Education, Vol. 4, no. 1, pp. 45–60.

Jones, G., Ravid, G. and Rafaela, S., 2001. Information overload and virtual public discourse boundaries. *In Proceedings of Eighth IFIP Conference on Human-Computer Interaction*, Tokyo, Japan.

Knock, N., 2001. Compensatory adaptation to a lean medium: An action research investigation of electronic communication in process involvement groups. IEEE Trans. on Prof. Communication, Vol. 44, Iss. 4, pp. 267–285.

Kuk, G., 2006. Strategic Interaction and Knowledge Sharing in the KDE Developer Mailing List. *Management Science*, Vol. 52, no. 7, pp. 1031–1042.

Lakhani K. and Hippel E., 2003. How open source software works: 'free' user to user assistance. *Research Policy*, Vol. 32, no. 6, pp. 923–943.

Lave J. and Wenger E., 1991. *Situated learning: Legitimate peripheral participation*. Cambridge University Press.

Lerner, J. and Tirole, J., 2002. Some simple economics of open source. *Journal of Ind. Economics*, 50, 2, pp. 197-234.

Lussier, S., 2004. New Tricks: How Open Souce Changed the Way My Team Works. *IEEE Software*, 21, 1, pp. 68-72.

Mockus, A., Fielding, T. and Herbsleb, D., 2002. Two Case Studies of Open Source Software Development: Apache and Mozilla. *ACM Trans. Software Eng. and Methodology*, Vol. 11, no. 3, pp. 309-346.

Nooy, W., Mrvar, A. and Batagelj, V., 2005. *Exploratory Network Analysis with Pajek*, Cambridge University Press, New York.

Open Source Initiative, 1999, Open Source Definition, [online]. Source Available from: http://www.opensource.org/osd.html [last accessed Jan. 14, 2009, last updated July 24, 2006]

Raymond, E., 1998. *The Cathedral and the Bazaar*, O'Reilly, Sebatopol, CA.

Rohde, M., Klamma, R., Jarke, M. and Wulf, V., 2007. Reality is our laboratory: communities of practice in applied computer science. *Behaviour & Information Technology*, Vol. 26, no. 1, pp. 81 – 94.

Sowe, S., Stamelos, I., and Angelis, L., 2006. Identifying knowledge brokers that yield software engineering knowledge in OSS projects. *Information and Software Technology*, Vol. 48, Iss. 11, pp. 1025-1033.

Weber, S., 2004. *The Success of Open Source*, Harvard University Press.

Wenger E., 1998. *Communities of practice: learning, meaning, and identity*, Cambridge: Cambridge University Press.