Analysis of activity in open-source communities using social network analysis techniques

María del Rocío Martínez-Torres*

Department of Business Administration and Marketing, Faculty of Tourism and Finance, University of Seville, Seville, Spain

The success of an open-source software project is closely linked to the successful organization and development of the underlying virtual community. In particular, participation is the most important mechanism by which the development of the project is supported. The main objective of this paper is to analyse the online participation in virtual communities using social network analysis techniques in order to obtain the main patterns of behaviour of users within communities. Several open-source communities related to Linux ports to embedded processors have been studied, obtaining a set of indicators by modelling them as a social network. Exploratory factor analysis has been used to extract the main dimensions related to the participation process. Participation inequality, hierarchy and the cohesion of the community constitute the main dimensions characterizing the participation mechanism within communities. Obtained results highlight the necessity of guiding the organization and development of the community to achieve successful target software.

Keywords: virtual communities, social network analysis, factor analysis, open-source software, online participation

1. Introduction

Open-source software (OSS) represent a model of software development in which the source code is available to users and can be distributed with few limitations on possible modifications and distribution by third parties. This term has been exactly defined by the open-source initiative (Open Source Initiative 1999) using 10 key requirements. In particular, OSS projects are developed and released under some sort of 'open-source' licence that allows inspection and reuse of the software's source code (Crowston and Scozzi 2002; Martínez-Torres, Toral, and Barrero 2010). One of the most important aspects of OSS projects is the fact that they are 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 (Dinh-Trong and Bieman 2005). 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). In many cases, the results are even more successful than proprietary counterparts, like in the case of web servers (Apache) or scripting languages (PHP), or at least represent a

*Email: rmtorres@us.es

serious competence to proprietary software, as in the case of operating systems (GNU/Linux), web browsers (Mozilla) or database management systems (MySQL). Another advantage of the supporting community is that it avoids the use of huge financial resources to put the software through extensive testing and Quality Assurance, like a proprietary vendor will do. Instead, the open-source projects have the community as a resource (Lakhani and Hippel 2003; Gruber and Henkel 2006). Finally, several studies claim that OSS is developed faster and cheaper, and the resulting systems are more reliable than proprietary software (Mockus, Fielding, and Herbsleb 2002).

The literature on OSS has focused on several topics related to successful OSS development (Dinh-Trong and Bieman 2005), motivation of programmers and developers (Bonaccorsi and Rossi 2003; Von Hippel and von Krogh 2003), the benefits of OSS (Kogut and Metiu 2000) and its implications for the public sector (Applewhite 2003), public domain licensing (Gambardella and May 2006) or its relation with open innovation (West and Lakhani 2008; Barge-Gil 2010). A complete taxonomy of OSS research can be found in Aksulu and Wade (2010) and Martínez-Torres and Diaz-Fernandez (2014). They classify open-source communities as part of OSS production and OSS diffusion. Although code contributions are usually produced by a small percentage of individuals that constitute the core team of user-developers, there are also hundreds or even thousands of participants who can choose their level of participation. 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 is also focused on OSS development, but instead of analysing the development in terms of the source code produced, it is focused on the social relationships among virtual community members. More specifically, the purpose of this paper is the identification of the main dimensions that facilitate the participation process, which is one of the basic mechanisms by which communities keep active and alive. Social network analysis (SNA) techniques have been used to extract several global features of communities that are then statistically processed to achieve this objective.

Communities of Linux ports to different processors and frameworks have been chosen as a case study. Data from 11 virtual communities associated to Linux ports have been analysed for seven years, leading to 77 social networks. These communities have been selected because Linux ports to non-x86 processor and frameworks are oriented to professionals and researchers more than to the general public like desktop Linux. That means that the participation mechanism is essential to survive, and they need to maintain a core group of developers that must be continuously re-occupied by new community members.

The rest of the paper is organized as follows. The following section analyses the role of virtual communities in the development of OSS projects using the notion of communities of practice as the theoretical background. After that, the methodology based on SNA techniques is presented. The data collection section details how data from the different communities have been extracted while the data analysis section shows the application of the proposed methodology and the obtained results. After the discussion section, conclusions are drawn.

2. Research framework

Virtual communities have been studied from the perspective of Communities of Practice (CoP) developed by Lave and Wenger (1991). This concept refers to the process of social learning that occurs when people have a common interest in some subject or problem, and decide to 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). CoPs are not formal structures. Instead, they are informal entities, which exist in the mind of their members, and are glued together by the connections

the members have with each other, and by their specific shared problems or areas of interest (Wenger and Snyder 2000; Ardichvili, Page, and Wentling 2003). When interactions take place using electronic media, these communities are often referred to as 'virtual communities' (Johnson 2001; Chairatana 2009; Tsang and Park 2013).

According to Rheingold (1993), virtual communities can be defined as a social relationship aggregation, facilitated by Internet-based technology, in which users communicate and build personal relationships. They allow the creation of weak links among geographically dispersed individuals who regularly participate in the community. A different perspective is provided by the definition of Preece (2001), which considers an online community as 'a group of people, who come together for a purpose online, and who are governed by norms and policies'. This definition encourages a balanced view of both social and technical issues, and it is widely applicable to a wide range of communities. For example, it applies to communities that exist only online as well as communities that also have physical presence. Despite subtle differences in focus, researchers agree on the use of 'cyberspace' as essential for the identification of virtual communities (Koh and Kim 2004). Examples of virtual communities can be found in fields like education (Gallardo, Barrero, Martinez-Torres, Toral, and Duran 2007; Martínez-Torres et al. 2010), software development (Toral, Martínez-Torres, and Barrero 2009a) and consumer behaviour (Shang, Chen, and Liao 2006).

Several prior studies suggest the suitability of CoPs as a Knowledge Management tool (Philips and Bonner 2000). CoPs represent an approach to knowledge management focused on knowledge and knowing in practice (Lave and Wenger 1991; Wenger 1998). They provide an environment for people to develop knowledge through interaction with others in an environment where knowledge is created, nurtured and sustained (Hildreth and Kimble 2002). Allee (2000) points out that the community of practice is an intrinsic condition for knowledge to exist, since it cannot be separated from the group that creates it, uses it and transforms it.

The process underlying the construction and nurturing of soft knowledge in CoPs is called legitimate peripheral participation (LPP) (Lave and Wenger 1991). LPP describes the process by which a newcomer is integrated into the community. In this process, new members learn how to function as a community member through participation, and acquire the language, values and norms of the community. Learning is gradually achieved as an individual moves from being a novice, gaining access to community practices to complete socialization and thus becoming an insider or full member of the community. For instance, OSS projects websites provide forums and mailing lists where participants and contributors can report software improvements, needs or bugs, and share and discuss solutions to posted messages. The other half of the duality is reification, which means giving concrete form to something that is abstract. It is the process underlying the construction of hard knowledge. Both processes are developed together in CoPs and they affect the way in which meaning is negotiated.

This paper is focused on the participation process, which will be modelled as a social network with arcs among nodes of community members. However, the LPP process in OSS communities is mediated by the informal structure of the 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 a concentration is called 'participation inequality' (Kuk 2006; Toral, Martínez-Torres, Barrero, and Cortés 2009), and it can be explained by the different user profiles of open-source communities.

Participation inequality allows the categorization of OSS community members into 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 a project for a long period of time and make significant contributions to the development and evolution of the system. Moderators and leaders are included in this group.
- *Active developers*: they regularly make contributions to the project.
- *Peripheral developers*: they occasionally contribute new features to an 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.

In this paper, the main dimensions contributing to the LPP process have been identified by considering the different user profiles. The 77 social networks extracted from the 11 Linux ports communities have been analysed using SNA techniques, but considering also the sub-networks of active and core developers.

3. Methodology

Mailing lists have been chosen because they allow the collective reflection and community discussions, and activities are not just confined to software development or coding alone (Sowe, Stamelos, and Angelis 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. Nevertheless, these kinds of data do not provide any information about the social structure of the community or about the relationships among authors. In this paper, social networks have been extracted from threads of discussion, and SNA techniques have been applied to characterize the participation mechanism (Stefanone and Gay 2008). 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 $E \subseteq V \times V$. Some network analysis methods are easier to understand when graphs are conceptualized as matrices, Equation (1).

$$M = (m_{i,j})_{n^*n} \quad \text{where } n = |V|, \quad m_{i,j} = \begin{cases} 1 & \text{if}(v_i, v_j) \in E, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

In the case of a valued graph, real-valued weight function $w(e)$ is defined on the set of edges, i.e. $w(e) = Ex\,\Re$, and the matrix is then defined as given by Equation (2).

$$m_{i,j} = \begin{cases} w(e) & \text{if}(v_i, v_j) \in E, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

In the context of threads of discussion, V represents all the authors posting messages and E represents the successive answers among authors inside a thread, which is the basic unit considered (Jones, Ravid, and Rafaela 2001). The use of discussion threads as the basic unit of analysis is 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; Toral,

Martínez-Torres, and Barrero 2009b). 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 be a directed graph, with the direction of the arc given by the flow of information between two authors, and a valued graph, as an author is able to participate several times inside a thread or can answer to the same authors in different threads. In this case, the values of arcs are actually represented as multiple lines.

Networks can be partitioned using some discrete characteristics of vertices. For instance, several classes of vertices can be obtained using the function $w(e)$, that is, the strength 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 into 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. However, 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 have actually used the concept of out-degree of a vertex, that is, the number of arcs a given node sends. Therefore, the average out-degree of all vertices could be used to measure the structural cohesion of a network independently of the network size.

Other important characteristics in SNA are centrality and centralization (Yun and Lee 2013). Centrality refers to positions of individual vertices within the network, whereas centralization characterizes an entire network. One approach to centrality and centralization is based on the simple idea that information may easily reach people who are central in a communication network. Hence, the simplest indicator of centrality is the number of its neighbours, which is his or her degree. The higher the degree of a vertex, the more sources of information it has at its disposal, the quicker information will reach the vertex, so the more central it is (Nooy, Mrvar, and Batagelj 2005). However, degree centrality measures might be criticized because they only take into account the immediate ties that a vertex has, or the ties of the vertex's neighbours, rather than indirect ties to all others. One vertex might be tied to a large number of others, but those others might be rather disconnected from the network as a whole. In a case like this, the vertex could be quite central, but only in a local neighbourhood. As a difference, closeness centrality approaches emphasize the distance of a vertex from all others in the network by focusing on the distance from each vertex to all others. The closeness centralization is an index defined for the whole network, and it is calculated as the variation in the closeness centrality of vertices divided by the maximum variation in closeness centrality scores possible in a network of the same size. In general, both degree and closeness centrality are based on the reachability of a person within a network. But none of them take into account how crucial is a person to the transmission of information through a network. This approach is based on the concept of betweenness, which rests on the idea that a person is more central if he or she is more important as an intermediary in the communication network. 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 (Nooy et al. 2005; Toral, Martínez Torres, and Barrero 2010). Indicators related to density and centrality have been used to identify the main characteristics of participation in OSS communities. They can be grouped together using an exploratory factor analysis, obtaining as a result the main dimensions involved in the participation mechanism.

4. Data collection

The case study is based on Linux ports to embedded processors. Linux is a PC-based operating system that has been developed as OSS along the structure of the UNIX operating system, and it is one of the most prominent examples of OSS projects. 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 a 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 many communities supporting each one of these Linux ports, and this is an excellent opportunity for analysing a big group of more or less 'homogeneous' communities.

Up to eleven virtual communities have been considered. They are listed in Table 1. Nine of them are Debian Linux ports to different processor architectures. The Debian Project is an association of individuals who have made it a common cause to create a free operating system called Debian GNU/Linux, or simply Debian for short (Wu, Klincewicz, and Miyazaki 2006; Mateos-Garcia and Steinmueller 2008). The other two virtual communities are specific Linux ports to ARM and PowerPC processors. They have been considered because of the special importance of these two families of processors.

Table 1: Virtual communities considered

	URL	Description
The ARM Linux Project (ARM)	http://www.arm.linux.org.uk/	ARM Linux is a port of the successful Linux kernel to ARM-processor-based machines
Debian port to ARM (D-ARM)	http://lists.debian.org/debian-arm/	ARM port for Debian GNU/Linux. Debian fully supports a port to little-endian ARM
Linux PPC port (PPC)	http://penguinppc.org/	PowerPC Linux is the Linux kernel running on a PowerPC processor
Debian port to PowerPC (D-PPC)	http://lists.debian.org/debian-powerpc/	PowerPC port of Debian GNU/Linux. The PowerPC architecture allows both 64-bit and 32-bit implementations
Debian port to m68 k (D-68 k)	http://lists.debian.org/debian-68 k/	Motorola 68 k port of Debian GNU/Linux. Debian currently runs on the 68020, 68030, 68040 and 68060 processors
Debian port to Alpha (D-Alpha)	http://lists.debian.org/debian-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)	http://lists.debian.org/debian-mips/	MIPS port of Debian GNU/Linux, able to run at both endiannesses
Debian port to BSD (D-BSD)	http://lists.debian.org/debian-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)	http://lists.debian.org/debian-hppa/	This is a port to Hewlett-Packard's PA-RISC architecture
Debian port to Hurd (D-HURD)	http://lists.debian.org/debian-hurd/	The GNU Hurd is a totally new operating system being put together by the GNU group
Debian port to SPARC (D-SPARC)	http://lists.debian.org/debian-sparc/	This port runs on the Sun SPARCstation series of workstations, as well as some of their successors in the sun4 architectures

Typically, messages stored in mailing lists are publicly available month by month and year by year. They can be sorted using different criteria like authors, messages, dates and threads of discussion (Figure 1).

For the purpose of this paper, it is more interesting to order messages by threads of discussion, as they follow the sequence of interactions among users. Figure 2 illustrates the flow diagram for data extraction.

Each community is accessed using its URL from Table 1. A separate social network is extracted for each year and community. For each case, a double processing is performed as illustrated in Figure 1. First, the alias and e-mail of those users who have posted messages through the year are extracted from the heading of each message. A table of pairs alias−e-mail is then built with the aim of merging together those users changing their alias but using the same e-mail, or those users with the same alias and slight variations in their e-mails. As a result of this first stage, a final list of community members is obtained and they will be the nodes of the social network. The second stage consists of analysing each thread of discussion during the whole year to establish the arcs among nodes of the network. Following the criterion defined in the methodology section, a user (node) answering to a thread of discussion is tied to all the users who previously posted messages to this thread.

5. Data analysis

Each community will be analysed during the period 2003−2009, 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 analysed. 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 point 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 from each social network:

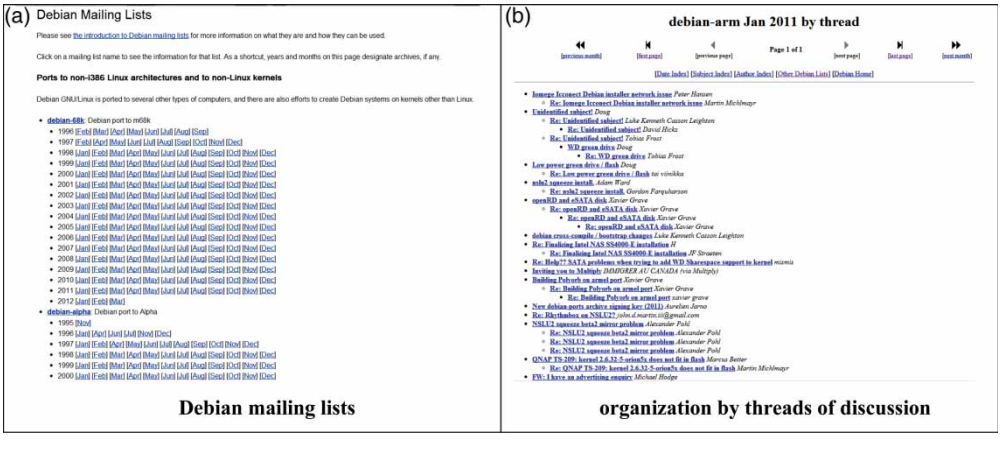

Figure 1: Mailing lists organization

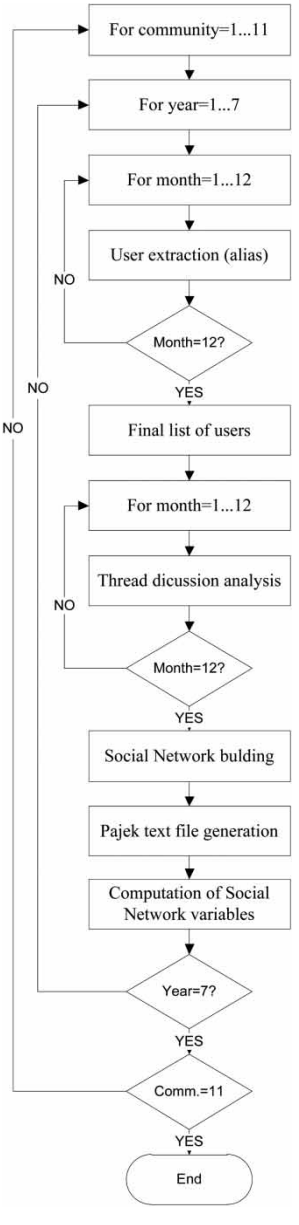

Figure 2: Flow diagram for data extraction

- *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 with respect to the whole community (V4) will be computed to consider the specific weight of this group.

- *Betweenness*: two values of betweenness will be considered: the betweeness of the sub-network of active developers (V5) and the betweenness of the sub-network of the core group of developers (V6).
- *Core developers*: the absolute value of core developers (V7) and their percentage with 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 values 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.

An example of the resulting social network is illustrated in Figure 3. Vertices represent community members and arcs represent the flow of information through threads of discussion. Arcs are valued with a value showing the number of interactions, although it has been omitted in Figure 1 for clarity purposes. External vertices (filled in white) correspond to peripheral members of the community, characterized by scarce interventions, while inner vertices (filled in red) correspond to active members. The direction of the arc is important because it shows the flow of knowledge. It means that a vertex with many inner arcs would be an information receptor while vertices with many outer arcs would be information providers. Vertices without arcs are passive observers. Betweenness centrality is determined by the ability of each vertex to go between two other vertices (Martínez-Torres 2013).

A factor analysis will be applied to extract the main dimensions related to online participation in virtual communities. Factor analysis attempts to identify underlying variables or factors, which can explain the pattern of correlations within a set of observed variables. Factor Analysis is a way to fit a model to multivariate data, estimating their interdependence (Rencher 2002; Martínez-Torres and Toral 2010). It addresses the problem of analysing the structure of interrelationships among a number of variables by defining a set of common underlying dimensions, the factors, which are not directly observable, segmenting a sample into relatively homogeneous segments (Toral and Martínez-Torres 2010). Factor analysis has been performed using the principal component method. The eigenvalues of the sample covariance matrix are shown in Table 2.

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. The Kaiser–Meyer–Olkin measure of sampling adequacy (0.711) and the Bartlett's test of sphericity

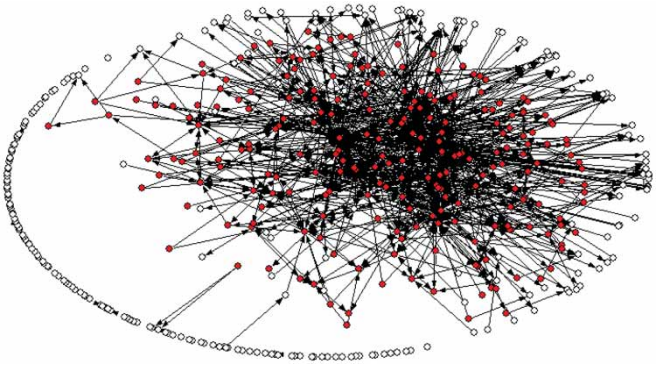

Figure 3: Social network of ARM Debian Linux mailing list community during 2008

Table 2: Total variance explained

Factor	Initial eigenvalues		
	Total	% 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	0.831	6.395	86.774
6	0.593	4.562	91.336
7	0.383	2.947	94.283
8	0.280	2.151	96.434
9	0.209	1.604	98.038
10	0.155	1.192	99.230
11	0.051	0.390	99.620
12	0.033	0.250	99.870
13	0.017	0.130	100.000

($\chi^2 = 969.35$, df = 78) suggest that the data set is appropriate for a factor analysis. 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 multiplication by an orthogonal matrix. The rotated loadings preserve the essential properties of the original 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 3 reports the rotated factor loadings with varimax rotation.

To extract the meaning of each factor, we move horizontally through Table 3, 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 (Rencher 2002). The association between variables and factors is highlighted in

Table 3: Rotated component matrix with varimax rotation

	Component		
	1	2	3
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

grey in Table 3. The resulting aggregation of variables leads to the following latent factors or dimensions detailed in Table 4. Notice that V9 cannot be clearly assigned to any factor.

On the other hand, factor scores are used to categorize the original sample, which can be approximated to one of the identified latent factors. An analysis of variance (ANOVA) has been performed to check the null hypothesis of equal population means. These null hypotheses have been rejected in all the cases with a significance value below 0.05, except for variable V9, which was removed from the analysis, Table 5. Using this categorization, the mean value of each variable per factor is detailed in Table 6. This information can be used in conjunction with factor loading for the factors' interpretation.

The first factor of Table 4 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 values of V10 and V11 confirm that the group of active and core developers is responsible for the majority of contributions.

The second factor of Table 4 is related to the hierarchy of the community. The core group plays an essential role for the continuity of the community and it must develop a brokerage role among contributors. A high proportion of active developers is required, because only a small fraction of them will become experts through the LPP process, joining the core group. In turn, the core group must perform an intermediation role to facilitate the process of becoming an expert.

The third factor of Table 4 is related to the cohesion of the community. Centrality (V5 and V6) and the structure of the communities (V8) are included in this factor. The negative value associated with V8 means that the core group should be just a small fraction of active developers, to guarantee a good coordination of the community.

The three obtained dimensions are graphically shown in Figure 4, while Figure 5 details the position of the considered Linux Debian communities in terms of the three obtained patterns of behaviour. It can be noticed that, except for two communities (Debian-hurd and Debian-mips), the rest of them do not exhibit pure behaviour as described by the obtained factor but a combination of the three of them. Most of the communities tend to achieve a community with high hierarchy and cohesion, which means that the core group exerts a meaningful influence over the rest of the community trying to attract as many active developers as possible.

Table 4: Identified factors

	Description	Loading
F1		
V1	Average out-degree (whole network)	0.929
V2	Standard deviation of out-degree values	0.821
V10	Average out-degree (active developers sub-network)	0.968
V11	Average out-degree (core developers sub-network)	0.963
F2		
V3	Number of active developers	0.699
V7	Number of core developers	0.923
V13	Number of brokers (core developers sub-network)	0.866
F3		
V5	Betweenness centrality (active developers sub-network)	0.721
V6	Betweenness centrality (core developers sub-network)	0.795
V8	Core developers/active developers	−0.794

Table 5: Statistical significance of ANOVA

	F	Sig.		*F*	Sig.
V1	20.990	.000	V8	3.850	.013
V2	10.711	.000	V9	1.005	.396
V3	9.922	.000	V10	22.213	.000
V4	7.473	.000	V11	24.521	.000
V5	9.293	.000	V12	8.222	.000
V6	16.563	.000	V13	16.813	.000
V7	23.041	.000			

Table 6: Mean values per factor of selected variables

	F1	F2	F3		F1	F2	F3
V1	16.41	8.85	5.82	V7	17.18	26.56	13.68
V2	77.14	35.57	32.25	V8	26.92	26.59	19.92
V3	73.18	104.44	77.31	V10	13.74	5.58	3.97
V4	15.97	23.67	23.59	V11	310.39	105.64	83.80
V5	0.08	0.07	0.12	V12	77.91	83.86	80.46
V6	0.34	0.31	0.53	V13	10.43	16.76	5.13

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 many users who are only interested in asking for information, but with no intention of becoming active contributors. Just a small fraction of visitors will become active contributors 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 LPP process can only be successful for a small fraction of users who decide to get involved in the community.
- The key role of a hierarchy controlled by the core group. The mission of the core group is not just participating but, above all, promoting the debate and participation and addressing

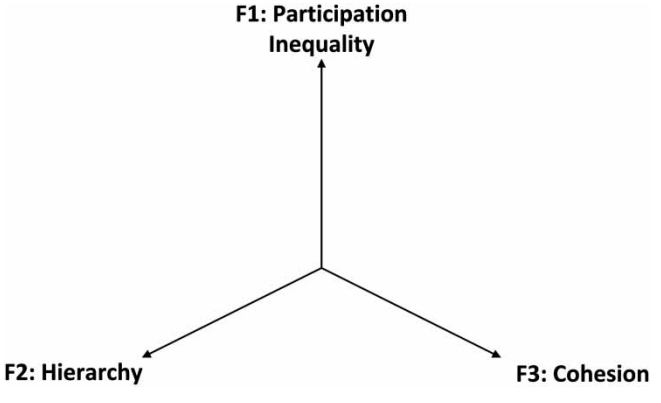

Figure 4: Interpretation of identified factors

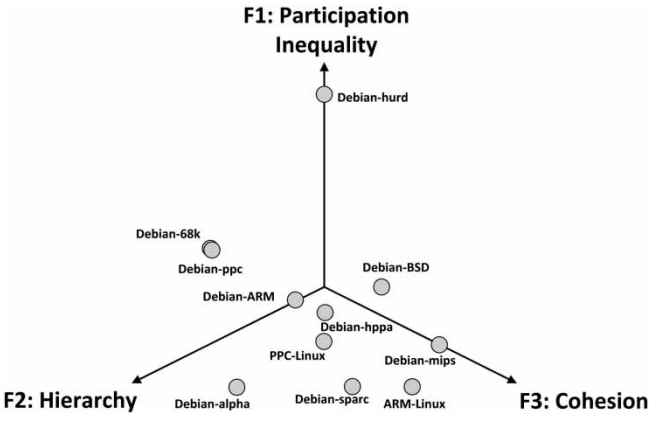

Figure 5: Patterns of behaviour of Linux ports communities

the future development of the underlying project. LPP process is only possible if there are experts spending time to solve other members' questions and facilitating their learning. They must also identify those active users who are ready to be part of their group.

- Finally, the cohesion of the network is supporting the mechanism of participation, necessary for 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.

6. Discussion

The emergence of information and communication technologies has led to new forms of communities of practice which make an intense use of electronic media. The starting point of these communities is the social character of learning, which depends on interactions among people and the negotiation of meanings (Intarakumnerd 2005). Participation has been highlighted as the basic mechanism for the development of the community (Lee, Park, and Song 2009). However, participation must be guided and that is the reason why virtual communities exhibit a certain structure with the core group at the centre of the community as the most important group. The core group is not only responsible for the majority of contributions, but must also perform a brokerage role among the rest of the community members to guarantee that developments follow the right direction and the LPP process is working properly. It should strive to create an environment and culture that fosters a sense of belonging in the community and mechanisms that encourage and enable newcomers to move towards the centre of the community through continual contributions (Yunwen and Kishida 2003). In this sense, the core group is responsible for maintaining the cohesion of the network, with the final objective of obtaining answers to the posted questions. Otherwise, new users can feel frustrated and decide to migrate to other communities.

One of the main advantages of the LPP mechanism is that community members do not occupy fixed positions over time. They can move through the different layers of the community as they progress in their experience. This fact provides a feedback mechanism over time to bring new users to the deeper layers of the community as some others can decide to abandon it or move to different communities. In the case of corporate-sponsored communities, these movements can be limited by the need of the company to retain a certain control or influence over the community (West and O'mahony 2008).

Finally, the layered structure facilitates the coordination mechanisms, making it easier to guide future developments. It is important to notice that hierarchies in OSS communities acquire the form of social hierarchies instead of being person-specific. Social hierarchies are informal and impersonal, and the patterns of cooperation in these social hierarchies can survive beyond the individuals that populate them. This fact explains how OSS projects can survive to the individuals by which they were originally created. In fact, the governance approach of the OSS project has significant consequences on the development and quality of the underlying software (Capra, Francalanci, and Merlo 2008). On one hand, open governance facilitates the LPP process, the incorporation of new members to the core group as well as new advances and research lines during the development of the projects. On the other hand and as a counterpart, excessive openness also may cause loss in the cohesion of the community and dispersion in the topics and issues that require new advances. Additionally, open governance approaches also require communication and coordination overhead and tangible additional effort. Obtained results suggest that most OSS communities tend to balance hierarchy and cohesion. One way of reaching such a balance is through the so-called community manager, which is an emergent profile in the field of OSS communities. The most fundamental responsibility of community managers is to bring people together, coordinating efforts between developers, ensuring that issues brought up on mailing lists are addressed fairly and linking the community strategy to the OSS project. The community manager is the bridge between the software and the core group of developers and the rest of the community. As part of his work, he must monitor the behaviour of the community, collecting, analysing and processing data and making decisions. The approach proposed in this paper based on SNA can help community managers to accomplish these tasks.

There also some limitations resulting from the methodology applied. The major limitation is that SNA techniques only consider the participation features of community members, that is, the quantity of posted messages and how each author is related to the rest of the community through threads of discussion. SNA does not consider the quality of posted messages. Such an analysis would require natural language processing techniques or text mining techniques, like latent semantic indexing or generative models (Toral, Martínez Torres, Barrero, and Arahal 2010). However, and even using these techniques, it is not easy to evaluate the quality of messages through computational algorithms. Natural language processing techniques rely on previous taxonomy of the field under study and a reduction in the high dimensionality of the feature space. They have been successfully used to determine the main topics within a certain field, but further analyses would be necessary to determine to what extent these topics are related with quality.

Another limitation related to the case study is the fact that collected communities exhibit low participation inequality compared to other OSS projects. This point can be explained because selected communities are oriented to professionals and practitioners of non-desktop Linux distributions, leading to smaller communities.

7. Conclusions

Communities are basically based on interactions among users, and participation is the basic mechanism promoting their development. This participation has been analysed using SNA 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 analysed using factor analysis. The obtained results reveal three main facilitators of the LPP process in OSS 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. The paper provides some important implications about the governance of OSS communities. First, a balance between

hierarchy and cohesion must be achieved, and this balance depends on the degree of openness of the project. Although openness leads to better results, it also requires a higher coordination effort of the core group. Many studies highlight the role of an emergent profile such as the community manager, responsible for monitoring the general behaviour of the community and deciding about the most appropriate governance style. Tools based on SNA like the one proposed in this paper can help community managers to monitor the participation features of users within the community. As a future work, the proposed techniques could be complemented with some natural language-processing techniques. Combining both of them, community managers could analyse not only the participation features but also the content of shared messages. More specifically, it could be studied to what extent a set of target topics agreed on between the community manager and the core group is aligned with the real topics discussed within the community.

Acknowledgement

This work has been supported by the Consejería de Economía, Innovación, Ciencia y Empleo (Research Project with reference P12-SEJ-328).

References

Aksulu, A., and Wade, M.R. (2010), 'A comprehensive review and synthesis of open source research', *Journal of the Association for Information Systems*, 11(11/12), 576–656.

Allee, V. (2000), 'Knowledge networks and communities of practice', *OD Practitioner*, 32(4), 1–15.

Applewhite, A. (2003), 'Should governments go open source?', *IEEE Software*, 20(4), 88–91.

Ardichvili, A., Page, V., and Wentling, T. (2003), 'Motivation and barriers to participation in virtual knowledge-sharing communities of practice', *Journal of Knowledge Management*, 7(1), 64–77.

Barge-Gil, A. (2010), 'Open, semi-open and closed innovators: towards an explanation of degree of openness', *Industry & Innovation*, 17(6), 577–607.

Bonaccorsi, A., and Rossi, C. (2003), 'Why open source software can succeed', *Research Policy*, 32, 1243–1258.

Capra, E., Francalanci, C., and Merlo, F. (2008), 'An empirical study on the relationship among software design quality, development effort, and governance in open source projects', *IEEE Transactions on Software Engineering*, 34(6), 765–782.

Chairatana, P.-A. (2009), 'Knowledge, innovation, and service system in latecoming Southeast Asia', *Asian Journal of Technology Innovation*, 17(1), 143–163.

Crowston, K., and Scozzi, B. (2002), 'Open source software projects as virtual organisations: competency rallying for software development', *IEE Proceedings – Software*, 149(1), 3–17.

Dinh-Trong, T.T., and Bieman, J.M. (2005), 'The FreeBSD Project: a replication case study of open source development', *IEEE Transactions on Software Engineering*, 31(6), 481–494.

Gallardo, S., Barrero, F., Martinez-Torres, M., Toral, S.L., and Duran, M.J. (2007), 'Addressing learner satisfaction outcomes in electronic instrumentation and measurement laboratory course organization', *IEEE Transactions on Education*, 50(2), 129–136.

Gambardella, A., and May, B.H. (2006), 'Proprietary versus public domain licensing of software and research products', *Research Policy*, 35, 875–892.

Gruber, M., and Henkel, J. (2006), 'New ventures based on open innovation – an empirical analysis of start-up firms in embedded Linux', *International Journal of Technology Management*, 33(4), 356–372.

Hildreth, P.M., and Kimble, C. (2002), 'The duality of knowledge', *Information Research*, 8(1), 1–17.

Intarakumnerd, P. (2005), 'The roles of intermediaries in clusters: the Thai experiences in high-tech and community-based clusters', *Asian Journal of Technology Innovation*, 13(2), 23–43.

Johnson, C.M. (2001), 'A survey of current research on online communities of practice', *Internet and Higher Education*, 4(1), 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 Transactions on Professional Communication*, 44(4), 267–285.

Kogut, B., and Metiu, A. (2000), 'The Emergence of E-Innovation: Insights from Open Source Software Development', Philadelphia, PA: Reginald H. Jones Center Working Paper.
Koh, J., and Kim, Y.-G. (2004), 'Knowledge sharing in virtual communities: an e-business perspective', *Expert Systems with Applications*, 26(2), 155–166.
Kuk, G. (2006), 'Strategic interaction and knowledge sharing in the KDE developer mailing list', *Management Science*, 52(7), 1031–1042.
Lakhani, K., and Hippel, E. (2003), 'How open source software works: 'free' user to user assistance', *Research Policy*, 32(6), 923–943.
Lave, J., and Wenger, E. (1991), *Situated Learning: Legitimate Peripheral Participation*, Cambridge, UK: Cambridge University Press.
Lee, Y.G., Park, S.H., and Song, Y.I. (2009), 'Which is better for a firm's financial performance: an externally oriented or inwardly oriented innovation strategy? An empirical study on Korean SMEs', *Asian Journal of Technology Innovation*, 17(1), 57–73.
Lussier, S. (2004), 'New tricks: how open source changed the way my team works', *IEEE Software*, 21(1), 68–72.
Martínez-Torres, M.R. (2013), 'Application of evolutionary computation techniques for the identification of innovators in open innovation communities', *Expert Systems with Applications*, 40(7), 2503–2510.
Martínez-Torres, M.R., and Diaz-Fernandez, M.C. (2014), 'Current issues and research trends on open-source software communities', *Technology Analysis & Strategic Management*, 26(1), 55–68.
Martínez-Torres, M.R., and Toral, S.L. (2010), 'Strategic group identification using evolutionary computation', *Expert Systems with Applications*, 37(7), 4948–4954.
Martínez-Torres, M.R., Toral, S.L., and Barrero, F. (2010), 'The role of Internet in the development of future software projects', *Internet Research*, 20(1), 72–86.
Mateos-Garcia, J., and Steinmueller, W.E. (2008), 'The institutions of open source software: examining the Debian community', *Information Economics and Policy*, 20, 333–344.
Mockus, A., Fielding, T., and Herbsleb, D. (2002), 'Two case studies of open source software development: Apache and Mozilla', *ACM Transactions on Software Engineering and Methodology*, 11(3), 309–346.
Nooy, W., Mrvar, A., and Batagelj, V. (2005), *Exploratory Network Analysis with Pajek*, New York: Cambridge University Press.
Open Source Initiative (1999), Open Source Definition [online], Source available from: http://www.opensource.org/osd.html [last accessed January 14, 2009, last updated July 24, 2006].
Philips, J., and Bonner, P.D. (2000), 'Motivation, knowledge transfer, and organizational forms', *Organization Science*, 11(5), 538–550.
Preece, J. (2001), 'Sociability and usability: twenty years of chatting online', *Behaviour and Information Technology*, 20(5), 347–356.
Rencher, A.C. (2002), *Methods of Multivariate Analysis*, 2nd ed., Wiley Series in Probability and Statistics, New York: John Wiley & Sons.
Rheingold, H. (1993), *The Virtual Community: Homesteading on the Electronic Frontier*, Reading, MA: Addison-Wesley.
Shang, R.-A., Chen, Y.-C., and Liao, H.-J. (2006), 'The value of participation in virtual consumer communities on brand loyalty', *Internet Research*, 16(4), 398–418.
Sowe, S., Stamelos, I., and Angelis, L. (2006), 'Identifying knowledge brokers that yield software engineering knowledge in OSS projects', *Information and Software Technology*, 48(11), 1025–1033.
Stefanone, M.A., and Gay, G. (2008), 'Structural reproduction of social networks in computer-mediated communication forums', *Behaviour & Information Technology*, 27(2), 97–106.
Toral, S.L., and Martínez Torres, M.R. (2010), 'International comparison of R&D investment by European, US and Japanese companies', *International Journal of Technology Management*, 49(1/2/3), 107–122.
Toral, S.L., Martínez-Torres, M.R., and Barrero, F. (2009a), 'Modelling mailing list behaviour in open source projects: the case of ARM embedded Linux', *Journal of Universal Computer Science*, 15(3), 648–664.
Toral, S.L., Martínez-Torres, M.R., and Barrero, F. (2009b), 'Virtual communities as a resource for the development of OSS projects: the case of Linux ports to embedded processors', *Behavior and Information Technology*, 28(5), 405–419.
Toral, S.L., Martínez-Torres, M.R., Barrero, F., and Cortés, F. (2009), 'An empirical study of the driving forces behind online communities', *Internet Research*, 19(4), 378–392.
Toral, S.L., Martínez Torres, M.R., and Barrero, F. (2010), 'Analysis of virtual communities supporting OSS projects using social network analysis', *Information and Software Technology*, 52(3), 296–303.
Toral, S.L., Martínez Torres, M.R., Barrero, F., and Arahal, M.R. (2010), 'Current paradigms in intelligent transportation systems', *IET Intelligent Transport Systems*, 4(3), 201–211.

Tsang, D., and Park, Y. (2013), 'How culture and government shape entrepreneurial innovation: the case of Korean and UK online gaming firms', *Asian Journal of Technology Innovation*, 21(2), 237–250.
Von Hippel, E., and von Krogh, G. (2003), 'Open source software and the "private-collective" innovation model: issues for organization science', *Organization Science*, 14(2), 209–223.
Wenger, E. (1998), *Communities of Practice: Learning, Meaning, and Identity*, Cambridge: Cambridge University Press.
Wenger, E.C., and Snyder, W.M. (2000), 'Communities of practice: the organizational frontier', *Harvard Business Review*, 78(1), 139–144.
West, J., and Lakhani, K.R. (2008), 'Getting clear about communities in open innovation', *Industry & Innovation*, 15(2), 223–231.
West, J., and O'mahony, S. (2008), 'The role of participation architecture in growing sponsored open source communities', *Industry & Innovation*, 15(2), 145–168.
Wu, Q., Klincewicz, K., and Miyazaki, K. (2006), 'Analysis of the open source software sector in China', *Asian Journal of Technology Innovation*, 14(2), 117–141.
Yun, S., and Lee, J. (2013), 'An innovation network analysis of science clusters in South Korea and Taiwan', *Asian Journal of Technology Innovation*, 21(2), 277–289.
Yunwen, Y., and Kishida, K. (2003), 'Toward an understanding of the motivation of open source software developers', in *Proceedings of the 25th International Conference on Software Engineering*, ICSE 03, Portland, OR, USA, pp. 419–429.