The periphery on stage: Functions, Properties and Dynamic Evolution of the Periphery in the Free/Open Source Software community

This version*: April, 2009*

Submitted to Research Policy

Francesco Rullani

DRUID & INO, Dept. of Innovation and Organizational Economics, Copenhagen Business School

Kilevej 14A, 2000 Frederiksberg, Denmark, Ph: +45 3815 2992; Fax: +45 3815 2540. Email: fr.ino@cbs.dk

Abstract

This paper focuses on the peripheral participants of the Free/Open Source Software community. It characterizes the periphery's properties and functions in the division of labor with the core, and shows that this division of labor is possible only if the core propagates to the periphery its "structures", i.e. social norms, production procedures, and vision of the community organization. A conceptual model based on the three concepts of "structure enactment", "negotiation of meanings" and "dissonance" is developed to describe the process by which the structures propagate toward the periphery and the role of possible moderators: periphery's atomization and invisibility.

Key Words*: free/libre/open source software, periphery, core, practice, inscription, artifact, dissonance, structure*

Acknowledgements: I am grateful to Alfonso Gambardella, Paul David, Paola Giuri, Salvatore Torrisi, Matteo Ploner, Carlo Carraro, and Pierluigi Sacco for insightful and inspiring discussions, Yuwei Li, and Lars Risan for their comments and help; Davide Consoli, Alexander Frenzel Baudisch, Marco Giarratana, Oliver Alexy, Jean-Michel Dalle, and Christopher Tucci for their valuable advice. I also thank Deborah Gray for her precious help in the copyediting of the paper. I want to thank the Free Software Users Group Pluto-Padova and the Venice Linux Users Group for their help. Thanks to the participants of the workshop "Socio-technical Dynamics in the Free/Libre Open Source Software (FLOSS) Social World", Helsinki (Finland) and to the DRUID summer conference 2008, Copenhagen (Denmark). This paper benefited from a grant in the framework of the Doctoral Program in Economics and Management, Sant'Anna School of Advanced Studies (Pisa, Italy) that allowed me to accept an invitation to spend five months at the Stanford Institute for Economic Policy Research as a Visiting Researcher of the Stanford (NSF) Project on the Economics of Open Source Software. I also gratefully acknowledge the financial support this work received from Fondazione IRI in the context of the program "Programma di perfezionamento all'estero in discipline manageriali".

1. Introduction

The aim of this paper is to develop a conceptual model able to account for one of the less studied collective actors of the Free/Open Source Software (FOSS)—the periphery of the community. Peripheral members, those who orbit around FOSS projects and contribute only sporadically with bug reports, suggestions, comments or "improvised solutions", are key for FOSS success (e.g. Raymond, 1998a). However, the literature has looked at their roles mainly from the point of view of the core developers, without trying to unfold their characteristics as a whole. The periphery has been treated mostly as the "*sparring partner*" of the core.

The contribution of this paper can then be found in the theoretical account of the periphery and its relationship with the core. The paper does that through a conceptual model connecting three elements: 1) the properties of the periphery and the consequent functions it can fulfill in the division of labor with the core, 2) the conditions necessary for those functions to be performed, and 3) the possible challenges the whole system faces due to the periphery peculiarities. Using this model, managers as well as researchers can elaborate strategies aimed at fostering the efficiency and efficacy of the division of labor not only for FOSS-related projects, but also for online communities in general.

In particular, the first passages of the paper are meant to characterize the periphery, its properties, and functions. In the division of labor between the two "tiers" of the organization of the FOSS community (Lee and Cole, 2003), the periphery implements those functions that hinge upon its characteristic of favoring quantity (the number of mobilized individuals) at the expense of the quality of the performed tasks. Bug reporting, monitoring members' rule compliance, providing improvised solutions to technical problems, signaling rule infringement, and providing the pool of individuals that can engage in a *legitimate peripheral participation* process (Lave and Wenger, 1991) that can fuel the core with new individuals, are all capabilities that rest on the large number of peripheral participants, rather than on the quality (in terms of time and effort) of their contributions. This in turn is the consequence of the self-organizing nature of the FOSS model, which needs to *dissipate* a great amount of resources to assure that the whole dynamic does not come to an end (David and Rullani, 2008; Lanzara and Morner, 2005).

As a further step, this paper argues that the peculiar division of labor described above is possible only if the periphery correctly perceives the core's *structures*, i.e. the authoritative configuration, the norms and the procedures supporting, but also created by, core developers' activities (Crowston

et al., 2005; Giddens, 1984). If there is a mismatch between the emergent core's structures and their peripheral members' perception, the latter's contribution will be much less effective, as will be the division of labor between the two social bodies that comprise the community. The core needs, then, to propagate its structures outward.

Following Orlikowski's argument on the use of technology in organization (Orlikowski, 2000), the paper then poses that the artifacts produced by the core are the vehicles through which this propagation takes place. This because the structures produced by the core's are inscribed in the artifacts it produces (from code to online discussions). Thanks to this inscription, when peripheral members face those artifacts, they are exposed to the core's structures, and relate to them in different ways, among which is the adoption and assimilation of the core's structures themselves.

To explain how this process works, I focus on a particular type of artifact: threaded discussions capturing developers' interaction online. This allows a clearer identification of the processes underpinning the argument, but what is said can be easily adapted, mutatis mutandis, to other kinds of artifacts produced by the FOSS community.

Given this, I combine Orlikowski's (2000) idea of "structures enactment" with Wenger's (1998a, 1998b) concept of "negotiation of meanings" to describe how peripheral members can be affected by the footprint of the core's structures carried by the artifacts. Consider first that the discussions undertaken by the developers are online, asynchronous, and public, so that peripheral individuals browsing a FOSS project's website are easily "exposed" to the *discourse* developed by the community. The discourse contained in the online messages has emerged as a result of the debates and activities of the core, and thus brings with it the traits of the structures the core has adopted. At the same time, it is also a collection of opinions, carrying problems and views that peripheral members may be unfamiliar with, ideas that may be in sharp contrasts to their own principles. When facing the discourse, then, peripheral individuals are exposed to a provocative debate, and pushed to form opinions on the topics considered crucial by the community, judging others' ideas but also challenging their own visions. The resulting process of confrontation between one's own "system of meanings" and that coming from the discourse is nothing other than Wenger's (1998a, 1998b) "negotiation of meanings". The outcome of that confrontation, as I will explain further in the paper, is that the core's structures inscribed in the discourse emerge in the everyday practice of the peripheral members, precisely as described by Orlikowski (2000) with the expression "structures enactment".

The second part of the paper examines each step of this process. It first highlights the importance of online communication in the FOSS community and the impact archived threaded discussions can

have on peripheral members' everyday practice. It then attempts to describe the events that might be taking place when a peripheral member faces these artifacts. The starting point is the observation that, when the structures conveyed by this process are not consistent with the individual's established sets of rules and beliefs, incoherence is created in the internal components of his or her identity, i.e. *dissonance* (Kuran, 1998). Individuals may reduce dissonance by internalizing the structures, i.e. reconstructing their internal beliefs and rules around new principles that embody the structures coming from the core. This mechanism allows the spreading of the core's structures to the periphery. However, this is only one possible outcome: revolting, leaving, or accumulating dissonance are also feasible answers to the contradiction. The process may remain in its "physiological" path only if the core is able to answer to peripheral members' dissonance filling the possible gap between its structures and the periphery's needs. However, two main obstacles may endanger this realignment. A first obstacle to this process is that individuals at the periphery are *invisible*, as they manifest their presence only when undertaking some action, i.e. by definition only seldom. The core thus can become aware of the status of the structure propagation process only from the few manifestations of the periphery. Moreover, the signals the periphery sends to the core are not only few, but also *atomized*. There is no overarching organization of the peripheral members, and each one has its own idiosyncratic answer to the core's stimuli. This transforms the many signals coming to the periphery into "noise", and undermines the core's perception of feedback on the structure propagation process.

In terms of its structure, the paper develops as follows. After discussing the importance of the periphery and identifying the contributions of the paper in section 2.1, in section 2.2 I identify what specific functions the periphery can accomplish better than the core on the basis of its own characteristics. In section 2.3 I argue that peripheral members' understanding of the core's *structures* (i.e. the technical procedures and social rules governing the coordination and participation of the core developers) is key for those functions to be realized. In section 2.4 and 2.5, I try to capture the main mechanisms behind the core's structure propagation using Orlikowski's (2000) concept of "enactment of structure" and comparing it to Wenger's (1998b) idea of "negotiation of meanings". The details of the conceptual model describing how the core propagates its structures to the periphery are developed in the second part of the paper (section 3). In section 3.1, I discuss the importance of the core's online discussions and its impact on peripheral members. In section 3.2, I introduce the concept of dissonance (Kuran, 1998) to describe how those artifacts can determine the structure propagation, and in section 3.3 I show how some of the properties typical of the periphery (namely atomization and invisibility) can become moderators of this

process. Finally, section 4 concludes.

2. The Periphery

2.1. Contribution: why focus on the periphery

This paper focuses on the free open source software (FOSS)[1] innovation model. This model of innovation is based on a community comprising software developers who collectively produce applications and operating systems in a self-organized manner. The software is then distributed using a peculiar kind of license that allows free modification and open redistribution by any individual other than the licensor. One of the characteristics of the model is that production is organized around a core group of a few developers surrounded by a large periphery of community members who individually produce few contributions —in some cases none—mostly of marginal value. For example, in the case of Apache, one of the most successful and diffused open source programs, Mockus et al. (2000) show that the great majority of changes to the code have been done by only a few core developers. A similar picture is presented by Lee and Cole (2003) with respect to Linux, the most famous FOSS operating system.

This disproportion in the number of people actually populating the periphery and those belonging to the core has been widely documented by the literature on FOSS (e.g. Kogut and Metiu, 2001; Mockus et al., 2000). From these studies, the literature has moved on to analyzing the process that individuals may go through when progressing from the periphery to the core (Jensen and Scacchi, 2005; O'Mahony and Ferraro, 2006; von Krogh et al., 2003). Other scholars focused on how the core organizes its activities (e.g. Giuri et al. 2008a; Giuri et al. 2008b). In other words, the core and/or the process to enter the core have always been at the center of the research. This has come at the expense of the research on the periphery's specific characteristics and on its relationship with the core, most often only defined in residual terms or along very specific dimensions (e.g. Crowston et al., 2006, Crowston and Howison, 2005; Muller, 2006b; Ngamkajornwiwat et al., 2008).[2] Of course, I am not suggesting that a discussion of the periphery is novel in the literature on the FOSS phenomenon. Almost every article on the matter recognizes that openness, which makes it possible for many more individuals than those actually developing the software to spot the bugs and report ideas and suggestions, is the key in the FOSS model. My argument here is that besides this recognition, only a few studies have focused on the many individuals in the background, and even fewer have tried to provide an analysis of their relationship with the community (e.g. Berdou, 2007;

[1] For the scope of the paper, Free/Libre Open Source Software (FLOSS), Free/Open Source Software (FOSS) and Open Source Software (OSS) are considered synonymous.
[2] In this context it is interesting to recall the study of nonusers, users, and developers of FOSS undertaken by Dahlander and Mckelvey (2005).

Lakhani, 2006; Zhang and Storck, 2001) despite the fact that the periphery is the necessary counterpart of the core: *the latter will not be able to function without the former*.

This claim is justified by the observation that it is duality between the core and the periphery that makes the FOSS enterprise possible. At the center of the community, the *quality overcomes the quantity* in terms of participation: the few core developers are very committed to the FOSS projects on which they work and perform the greater part of the connected activities. On the contrary, the periphery is where the *low quality* of the participation, in terms of both the typology of the performed tasks and the participants' time and energy spent on them, *is counterbalanced by the quantity*. This generates a natural division of labor between the two social bodies, and it is the main reason why the core would not be able to perform the very specific functions of the periphery. Singh et al. (2007) gives empirical soundness to this claim showing that for a project to be more efficient and progress faster, both a clearly identifiable group of core developers *and* an "alive" periphery are needed. The conclusion is that the features of this undeniably blurred and fuzzy agglomeration of peripheral individuals "orbiting" FOSS-related projects, initiatives, and discussions are key for the FOSS innovation model–and for any distributed innovation model (Lakhani and Panetta, 2007)—to be fully realized, and should then be moved "onstage", at the center of the research.

This article develops a conceptual model aimed at describing the role of the periphery and the core in their division of labor, identifying what conditions make this division of labor possible, uncovering what processes realize these conditions, and flagging the properties of the periphery that moderate them. In doing so, particular emphasis is placed on the virtual environment in which the FOSS community is immersed. This allows the investigation of the specific characteristics of the periphery of the FOSS community, but also allows the generalization of the results, relating to all the other communities whose members use the virtual environment to carry on their joint activities. Thus, this analysis can uncover the role that the periphery has in online communities, the features of the division of labor with the core, and what threats are faced by the cooperation between the two social bodies. The model can then be useful to online community managers, as well as to scholars focused on virtual environments, as a first basis to elaborate strategies able to leverage the mass of peripheral individuals and establish a productive division of labor between them and the core. Being aware of the mechanisms behind peripherality is a crucial task for managers and scholars, because, as Brown (2004, p. 150) argues, "It is on the periphery that the next phenomenon […] might already be […]. Managers need to pay close attention to avoid being caught by surprise."

2.2. The functions of the periphery

The periphery of the FOSS community can be thought of as the set of actors marginally involved in the discussions, projects, and actions relative to community itself, but who are nevertheless interested in those activities, search for related information, and use the software produced by the community. They are more than simple users; they browse the community archives, observe its activities and contribute by signaling bugs, sporadically by sending patches, sometimes by participating marginally in the discussions of the community, but they are not in any other way actively involved in the community. In a word, they can be considered "lurkers", i.e. observers who exhibit only a limited level of activity.

Operationally, one can refer to the definition of the periphery provided by Crowston et al. (2006). The authors assign individuals to the core or the periphery according to their level of activity with respect to specific operations (e.g. being or not formally registered as developer, contributing above or below a certain threshold to the bug fixing process, being part or not of a clique of highly interacting participants). Similarly, Zhang and Storck (2001) examine the number of messages sent to the community forum and identify peripheral members as those who posted less than 30 messages in the sample period.[3] Theoretically, Wenger's (1998b) conceptualization of communities, which will inform the rest of the paper, can be used to define the periphery as the "cloud" of individuals *orbiting* developers, opinion leaders, and all the others who are deeply engaged in the community activity, and comprise the core of the community.[4]

An introduction to the functions of such periphery can be given by referring directly to the peculiar features of the FOSS production model. The first striking property of this model is that, when considering the community as a whole, it does not seem to efficiently employ its resources (i.e. individuals' energies, effort and time). Actually, it does not allocate most of the available resources, and most of the time it "wastes" what has been mobilized. David and Rullani (2008) show that a huge mass of developers has to be reached by the FOSS production model (at least to the level of registering to a development platform such as SourceForge.net) in order for just a few of them to actually become project members or to launch new FOSS projects. Defining a scale of three levels of activity (not contributing or doing so marginally, joining existing projects, and founding new projects) the authors observe that a move upward by one level implies the shrinking by one order of magnitude of the number of mobilized individuals. Moreover, even when dealing with the most active individuals, contribution does not last long after they have launched their first project in the

[3] This definition is essentially different from those that identify peripheral members not on the basis of the *level* of their activity, but on the *typology*. Berdou (2007), for example, differentiates members according to the degree that their activities belong to core tasks (e.g. coding) and non-core tasks (such as documentation or translation).

early months of their participation in this virtual environment. Most resources are thus dissipated. Notice that in a process where individuals self-select into tasks (Langlois and Garzarelli, 2008) and where there is no overall direction able to guarantee the matching between what is needed and what individuals pursue, this is an expected outcome.

Moreover, the internal order of a self-organizing system (and thus its capability to produce) is not kept by simply imposing a hierarchical structure. In the FOSS model, authority and leadership certainly exist, but they have to be continually renewed and legitimized through specific social processes (Dahlander and O'Mahony, 2008; Mateos-Garcia and Steinmueller, 2003; Muller, 2006b, 2006c; O'Mahony and Ferraro, 2007). These activities not only "cost" developers' energy, time, and attention, but sometimes they are also ineffective, and can result in further dispersion of resources through conflicts, defections, and the like. Notice that, even if technological, legal, and social rules, processes, and constraints can be ameliorated, the unfavorable rate between productive and available resources is again unavoidable, as it springs from the very characteristics of openness and self-organization of the FOSS itself. The FOSS model of innovation is inherently *dissipative* (David and Rullani, 2008; Lanzara and Morner, 2005).[5]

Even in this situation, however, a minority of the mobilized resources is actually used and transformed into the engine of the community. These few individuals are the drivers of the overall activity, creating new projects, joining, managing and coordinating existing ones, and carrying on most of the actions needed to give life and productive capabilities to the community.

As a first conclusion, it is possible to state that for such a system to survive the available resources must be much more than those actually used. More specifically, the huge number of individuals orbiting the core is essential to guarantee that at least a few will engage in the progression that will lead them into the core (i.e. into the process named 'legitimate peripheral participation' by Lave and Wenger, 1991). A wide periphery is then the condition sine qua non for the existence of the core, as the latter needs to be able to draw from the former enough resources to keep its production and social processes alive.

Going further with the analysis, it is possible to see that this disproportion in the number of individuals composing the two groups is compensated by the level of engagement they have in the community. As said, the core is composed by far less individuals than the periphery, but those

[4] A deeper discussion of Wenger's (1998b) and many others' definitions of periphery can be found in Lakhani (2006).
[5] The discussion on the features (as well as the fallacies) of this analogy with dissipative systems or structures (Prigogine and Stengers, 1984) is much wider than the scope of this paper. In this case, the term dissipation is meant to indicate only the necessity for the process to "waste" a certain amount of the mobilized resources to fuel the dynamics that keeps it alive and productive. As David and Rullani (2008) also state, the reader can refer to MacIntosh and MacLean (1999) for further discussion of the analogy in organization studies.

individuals represent the engine of all the community activities. In the periphery, the relationship between quality and quantity is the opposite: there are a huge number of individuals that are undertaking activities much less engaging, and—taken each one by itself—are also less relevant. This difference is again given by the nature of the periphery described above: it is a huge pool full of heterogeneous resources (Lakhani et al., 2006) that, to be mobilized, need to find their own idiosyncratic way into the system in a self-organizing manner. Thus, the activities peripheral individuals can perform *as periphery* inevitably belong to the set of low-profile, low-cost, and "improvised" activities, otherwise the effort to match the task requirements would overwhelm the already minimal incentive and commitment. In the FOSS, however, these activities turn out to be crucial. The openness of the code, i.e. its modularity and its easily accessible structure (MacCormak et al., 2006), makes it possible to realize a precise division of labor between the core and the periphery that increases the value of the contributions coming from the external regions of the community. Individuals at the periphery, even if minimally active, can engage in many micro-tasks at very low cost. Finding the bugs in the code is a very good example of this kind of task (Kogut and Metiu, 2001). The large number of people assures that the overall level of activity is high even if each individual performs only very simple and low-intensity tasks. Additionally, the higher heterogeneity of the individuals populating the periphery (Demazière et al., 2007a, 2007b) also assures that the scope of their needs and of their actions will be broad enough to guarantee the coverage of many features of the software (Raymond, 1998a; Bessen, 2006). This process has been exemplified by the famous Linus' Law reported by Raymond (1998a): "Given enough eyeballs, all bugs are shallow". Thus, the resources not mobilized into the process that direct them toward the core are nevertheless helpful as such, i.e. as peripheral resources, precisely because they can perform the tasks suitable for their peculiarities better than the core.

This process is however more complex and subtle than what the previous simple formulation suggests. Lakhani and von Hippel (2008) have shown that the development of one of the most important FOSS applications, PostgreSQL, is not performed by the core in a "pneumatic vacuum". On the contrary, the continual "interference" of more peripheral members is crucial to stimulate the process and provide solutions. The likelihood that a solution to a specific and well defined problem would come from the periphery is high because the periphery itself is composed of a plethora of individuals having heterogeneous knowledge and skills who are free from the mind-set built during the first phases of the problem formulation that could lock-in the problem-solving activity of the core (Lakhani et al., 2006). In other words, the periphery can provide the core not only with a large workforce able to perform activities such as bug reports, patches, production, or useful feature identification; it is also the place where out-of-the-box "improvised"—but crucial—solutions are

created in a distributed manner. As an example, consider the following description of this process in the context of "broadcast search".[6]

"A firm […broadcasted] problem […] was solved […] by a scientist with a Ph.D. in protein crystallography [who] applied common knowledge from crystallography to toxicology" (Lakhani et al., 2006: 10). *"This effect may be due to the ability of 'outsiders' from relatively distant fields to see problems with fresh eyes"* (Lakhani et al., 2006, p. 12).

Besides providing the pool of future resources, the mass of beta testers and debuggers, and the providers of improvised solutions, the periphery also has another role: it assures the functioning of the social practices of the community. The periphery participates not only in the technical processes of production, but also in the construction, replication, and preservation of the community beliefs and ideas. This fundamental role can be better understood by referring to the mechanisms at work to assure the enforcement of the rules of the community, both formal (e.g., the OSS licenses) and informal (i.e. the 'ownership' the leader of a project has on its development, Raymond, 1998b). The General Public License (GPL), for example, is a legal rule that erases the possibility of appropriating the code of the community, and coordinates developers' behaviors around cooperative strategies (Gambardella and Hall, 2006). However, the mere existence of the rules is not enough to assure individuals will comply with them. An enforcing mechanism is necessary to make agents internalize the likelihood of being caught infringing a rule and the cost of the relative punishment. The possibility to enforce the GPL—and thus the possibility for the GPL to have a function at all—is related to the public debate where developers can point out fraudulent behaviors.

"In the eyes of both legal scholars and informants, the GPL's strength stems not necessarily from its legality, but from the public collective opinion of community members. Informants also stressed that the primary vehicle by which they could enforce their license terms was by identifying and critiquing violations on on-line mailing lists and bulletin boards" (O'Mahony, 2003, p. 1189).

The *public collective opinion* conveyed by *online mailing lists and bulletin boards* plays a crucial role in the process. The opinions on misuse or infringement of the GPL are discussed publicly, and stored in the messages sent on the Internet. Another quote from O'Mahony (2003) shows how the periphery can have a crucial role in this:

"Reports of source code violations often come from customers of [copyright infringing] vendors who post to a community list or report the infraction to the copyright holder. We had three people writing to us saying, "My company bought this product from them, because we need this […] tool

[6] When applying broadcast search, firms' search processes do not take place inside the firm, but are developed broadcasting the problem to be solved to a wider audience of mostly unknown "outsiders". The provision of a workable solution by an "outsider" is usually compensated by a prize. See Lakhani et al. (2006) for more details on this model.

for the work we were doing, and we discovered that it looks like it is based on [your software], and darn it they didn't give us the source code (Informant, Non-Profit Foundation)' " (O'Mahony, 2003, p. 1187).

Monitoring and spreading information on copyright infringement—as well as any other behavior not respecting the rules of the community—is an activity that the periphery is in the position of doing effectively. Fallacies in others' code or behavior can easily be spotted by a large community of observers that is much larger than that of the few protagonists of the development process.

Moreover, as is the case for technical solutions, improvised stimuli relative to the social environment and culture of the community can once again come from the periphery. Elliott and Scacchi (2003) report a case in which a discussion of the basic values the FOSS community (namely, the lack of freedom attached to the use of non-FOSS code) is triggered precisely by an outsider.[7] The periphery is crucial for social control, rules compliance, and construction of the ethos at the foundation of the community activities.

Summarizing, the periphery has a crucial role in the FOSS model of innovation because:

A. it constitutes the pool from which the core draws the resources it needs to function. The progression from the periphery to the core of some developers is what assures the continual provision of new skills, energies, and ideas to the core;

B. it undertakes a series of technical tasks (such as bug reports and improvised solutions) that can help the core to improve and develop the code, enhancing its productivity; and

C. it constitutes the main devices through which social control (in form of monitoring or questioning members' rule compliance) is undertaken.

These functions are specific to the periphery as they derive directly from the properties described above, and in particular from the unbalanced ratio between the quantity and the quality of peripheral members' participation. The core, based on the opposite balance, is structurally unable to perform those functions at the same level of effectiveness. This process is represented at the bottom of Figure 1.

2.3. The importance of structure propagation from the core to the periphery

This section tries to move a step forward asking "What are the conditions for the previous process of division of labor to be fully realized?"

2.3.1. The starting point: mutual knowledge and common language

[7] I will report the details of this case more precisely later in the text.

FOSS innovation is based on the very difficult balance between cooperation—which requires the construction of a bridge between the participants allowing for mutual comprehension— and Computer Mediated Communication—where instead contextual information is shared only for a small part and there is no physical presence (e.g. Kock, 2004). As Cramton (2001, p. 349) reports "In the communications literature, mutual knowledge is considered to be a precondition for effective communication and the performance of cooperative work.". The same necessity is highlighted by Lazear (1999) in the context of team production. The author argues that a common language is the *condition sine qua non* allowing heterogeneous teams to gain from the diversity of their members.

In the context of FOSS, this commonality seems to be a necessary but not sufficient condition. As a matter of fact, Giuri et al. (2008a) show that the overlapping of team members' skills is not a crucial variable to explain FOSS projects productivity: its effect on the performance is unclear, while skill diversity is much more important. The authors argue that this is due to the fact that members of FOSS projects already share a wide knowledge basis and speak the same language, as they belong to the same 'hacker culture' and have skill sets that, even if differentiated in their articulation, all belong to a specific set of abilities (e.g. programming software or working in teams).

Even if no coordination is possible without a minimum level of mutual knowledge and common language, cooperation in the production of complex products seems to require more than just communication (see for example Zirpoli et al., 2009).

2.3.2. A step forward: structures

Crowston et al.'s (2005) operationalization of Giddens' *structuration theory* (Giddens, 1984) in the FOSS context can help in defining what the other elements are that need to be shared by the members of the community to create an effective division of labor. In general, structures can be conceived as the constraints and resources, or enabling factors, which shape individuals' actions in the societies they live. Structures, however, are not a top-down construct imposed from the social level onto the individual level. They emerge through individuals' interaction, and are in this sense an emergent property of collective processes. Individuals retain a certain level of agency, i.e. of the possibility to choose a different path of action in response to the structures they face. Crowston et al. (2005) take this concept and frame it into the FOSS discussion stating that individuals participating in the production of code create, maintain, and continually renew a set of *structures*, defined as follows:

"*By structure, we mean the rules and resources that influence, guide or justify individual action.... We specifically consider three kinds of rules and resources…*

1. interpretive schemes *that create structures of signification,*

2. authoritative and allocative resources *that create structures of domination, and*

3. norms *that create structures of legitimation*", (Crowston et al., 2005, p. 10, emphasis added).

These processes are those that need to be shared by the core and the peripheral individuals to make the latter effectively participate in the production process, while, at the same time not necessarily assimilating them into the core. Thus, our analysis needs to focus on the specific structures, such as *social rules* and *technical procedures* and their interpretations and representations by the agents, aimed at *guiding* the social and technical processes constituting the activity of the community.

As a first step, consider that given the difference in their roles and in the mutual relationship, the common set of structures needed for the periphery and the core to effectively cooperate are, most of the time, produced *by* and *in* the core, and then propagated to the periphery (Figure 1 provides a representation of this point). This gives the core the obvious opportunity to create structures consistent to its members' specific purposes and visions. However the consequent unilateral construction of structures may create the inability to reach the peripheral individuals and to effectively propagate the structures. This risk is paramount, as the commonality of structures is crucial for the FOSS project's capability to produce and develop. When the core fails in propagating its structures, each of the functions described above can be seriously underdeveloped. Consider for example that:

a. without shared *social rules*, i.e. of the '**structures of legitimation**', there will be no widespread monitoring, no detection, and no signaling of infringements (e.g. of the GPL terms) simply because the social rules to be enforced will not be shared by the periphery.

b. without shared *production procedures*, i.e. '**structures of signification**'[8] improvised solutions and micro-distributed processes of innovation (Lakhani and von Hippel, 2008) will be far less likely and difficult to combine into a coherent productive effort. If the peripheral individuals do not share the same idea of how the product should be developed and do not have in mind how these principles are translated into the product structure, it is very difficult for them to intercept problems they can easily implement solutions for. Even if they can match the problem to the knowledge they

[8] The mechanism described by Crowston et al. (2005) is based on the idea that individuals' mental models have to be shared in order for them to agree on the interpretation of their activity and of their cooperation. This idea is close to the concept of *cognitive distance* reported by Muller (2006c). When cognitive distance is low, i.e. cognitive models of agents are close one another, knowledge transfer became easier as well as sharing of common interests and incentives. The effects are consistent with those reported here.

have, they may fail to see it as the problem, and the route to the solution can be framed in a way unfamiliar to them.

c. along the same lines, without a shared *enactment of the formal and informal organization* of the community, its roles and the distribution of the control on different resource, i.e. '**structures of domination'**, the processes attracting peripheral individuals into the core (such as legitimate peripheral participation, [Lave and Wenger, 1991]) can become a long apprenticeship, where barriers to entry are higher and the likelihood of dropouts increases. Not being able to construct and interpret their experience of the community environment in the proper way (i.e. not being able to recognize and adopt the behavioral part of the "joining script"; von Krogh et al., 2003), produces the risk that newbies' participation will be less targeted toward the procedures that the core really care about and need to implement. The resulting mismatches and conflicts are likely to distract individuals' energies and attention from the productive processes and increase 'noise'.[9]

It is possible to recognize some of the passages discussed above in the case of Netscape's first attempt to release its browser as open source. As Bonaccorsi and Rossi (2005) state:

"*When, in 1998, the code of the Mozilla Web browser was released to the Open Source community under a non copyleft license, the Netscape Public License (NPL), it was able to attract very few contributions*" (Bonaccorsi and Rossi, 2005, p. 23)

This because, as Gambardella and Hall (2006) show, the choice of the initial license is crucial to determine the development of a specific software under the FOSS domain, and Netscape failed to act strategically in this sense. De Laat (2005) explains the reaction of the open source community – constituting the pool of resources Netscape's project was planning to draw from—very vividly:

"This [NPL] license proposal was 'beta-tested' in public, via a special web-site. Many hackers were enraged, especially by the special rights Netscape reserved for themselves" (de Laat, 2005, p. 1527).

In other words, there was a mismatching. On one side, the structures of domination and legitimation the core of the project had created. On the other side, the structures as perceived and applied by the periphery the project was trying to reach and attract resources from. Many of the peripheral members (i.e. developers possibly interested in the project) simply left. Others expressed their anger during the 'beta-testing' phase and in other forums and mailing lists, giving the community a clear signal about what was going wrong.

[9] For example, von Krogh et al. (2003) report that "there are implicit, but nevertheless important joining scripts in the Freenet project … . the developer favors hand-on solutions to technical problem, and … technical knowledge in the form of software code submission matters more than signaling of interest and experience", (von Krogh et al., 2003, p.

As a consequence, the company changed its license scheme, opting for a license more in line with the expectations of the community in terms of the "distribution of the power" over the development process. As Bonaccorsi and Rossi (2005) notice:

"This forced Netscape to add a copyleft license, the Mozilla Public License, to the NPL… in order to generate the feedbacks that allowed to release the first version of Mozilla in 2002", (Bonaccorsi and Rossi, 2005, p. 23).

That was not the only problem Netscape had to face. As Augustin (1999) argues, Netscape's idea of open source development's procedures, i.e. structures of signification, also mismatched the expectations of the periphery. The author states:

"*Even though Mozilla code was constantly open and available, there was never a release. In one year, the code was never stamped with a number, called "alpha", and never widely announced as a release. You can't go that long without calling something a release. People lose interest. Mindshare wanes. Developers begin to lose hope*" (Augustin, 1999, online).

Even more important, was the lack of an "'*architecture for participation' that promotes ease of understanding by limiting module size and ease of contributing by minimizing the propagation of design changes*" (MacCormack et al., 2006, p. 1028).[10]

The proof of such argument can be found in the increase of the number of Mozilla contributors after the company recognized the mistake and redesigned the code architecture (MacCormack et al., 2006).

Summarizing, the *structures of signification* as well as *domination* and *legitimation* the Netscape core developers were broadcasting into the periphery were in contrast with the peripheral members' own structures. The contradiction reached a point that many individuals did not participate in the process and some of them voiced their complaints. Cooperation became impossible and the project was doomed to fail. Consider how Augustin (1999) describes the situation at that time:

"Mozilla never achieved the success of the Open Source projects that inspired it. The contributor base has remained largely Netscape employees. The 'Open Source Community' never really embraced Mozilla and the project has stalled" (Augustin, 1999, online).

However, when the company recognized the mismatching between the structures the periphery was enacting and those shaping the core's activities, it adapted the latter to the former, regaining appeal and momentum.

1229).

[10] On this, the reader should also see Baldwin and Clark, 2006.

2.4. Filling the gap between core and periphery: structures propagation through artifacts

Having shown that structures propagation from the core to the periphery is a key factor allowing an effective division of labor between the two social bodies, the question relative to the process realizing such propagation becomes crucial.

First, consider that structures are not abstract mechanisms existing independently of human action. When individuals become part of a collective, they follow the configuration of constraints and resources they find in the social environment; in that very moment they realize, perform, or *enact*, the structures permeating that environment, and build the premises for their enforcement and reproduction. Every time a core developer mobilizes a certain resource or respects the limits imposed by a specific constraint, the structures underpinning those resources and constraints are in a certain sense "brought to life". In other words, core developers' practice-based interaction, well known and studied in the literature (Cohendet et al., 2001; Lin, 2003a, 2003b, 2004a, 2004b), is the anvil where structures are forged. As a consequence, structures such as social rules, technical procedures, and the related representations and interpretations, do not exist independently of the action, but are *enacted* by means of the practices undertaken during core developers' interaction.

Second, consider that artifacts are crucial in the construction and propagation of structures. When artifacts are collectively produced, the structures emerging during their production process are *inscribed* into the artifacts themselves, i.e. the artifacts' inner configuration contain and reflect the structures emerging during their construction. For example, Lanzara and Morner (2005) show how the organizational features of the FOSS production process are inscribed into the threaded messages circulating in the projects' mailing lists and stored in online public archives.

"*The developer mailing lists are electronic communication artifacts inscribing software-based protocols and procedures that allow specific interactions while prohibiting others, make possible specific ways of developing software jointly, and enact specific models of organizing and knowing*", (Lanzara and Morner, 2005, p. 73).

The organization of the debate in threads, the constellation of different opinions and opinion leaders emerging during the debate, and the way the interaction develops, all reflect the structures emerging during the discussion.

The consequence of the two mechanisms described above is that, being that the structures inscribed into the artifacts' configuration, **the artifacts become *vehicles* for the structures**. When individuals other than the artifacts' creators use the artifacts in their everyday practice, the objects' configurations ease certain actions and prevent some other actions, give room to certain opinions and restrict the role of others, and report a certain vision of the world while hiding other

representations. Orlikowski (1992; 2000) effectively puts forward this idea in the context of the relationship between technological artifacts and their users. She argues:

"*While a technology can be seen to have been constructed with particular materials and inscribed with developers' assumptions and knowledge about the world at a point in time… it is only when this technology is used in recurrent social practices that it can be said to structure users' actions*", (Orlikowski, 2000, p. 408).

When using the artifacts, then, the users enact the inscribed structures, so that the structures produced by the core reemerge during the users' use of the artifacts. This process creates a bridge between the core and the periphery across which it is actually possible to transfer the structures, *even if the latter does not participate* in the practices of the former. If the core and the periphery could engage in a dialog on equal levels, and through that dialog create a shared set of structures, this process would be realized following a route that has already been widely studied in the literature of communities of practice (Lave and Wenger, 1991; Lin, 2004a; Muller, 2006a; Wenger, 1998a). However, the periphery does not interact with the core from the same level. Peripheral members are not usually engaged in the core's activity, and when they are, they participate only marginally. This would make a transfer of structures based on individuals' direct interaction impossible. The above shows that even without direct interaction, the core's structures can be propagated to the periphery through their inscription into the core's artifacts.

Recall the previous description Lanzara and Morner (2005) provide of the process of inscription in threaded mailing list messages: The core produces mailing list discussions (together with other artifacts) and stores them in publicly accessible archives. The messages reflect the structures emerging in the core, as they are inscribed during the discussion in the configuration of the threads and in the web of opinions conveyed by the messages. Peripheral members access the repositories of the mailing lists searching for information or simply looking to be updated on the development process. The web of opinions, visions, and ideas that peripheral individuals face forms a web of constrains (e.g. through discussion delegitimizing certain set of arguments) and enabling factors (e.g. through opinions supporting other lines of argument) reproducing the inscribed structures. When peripheral members use the artifacts, i.e. they read the discussion threads searching for the information they need, they form opinions, contrast others' ideas or modify their own visions according to what others say, and act accordingly. Through this use of the messages in their everyday practice, they enact the inscribed set of structures (Orlikowski, 2000). Thus, structures are "brought back to life" into peripheral individuals' practices even if peripheral individuals do not participate actively in the core's activities.

Notice that what has been said for the mailing lists messages is also true for other typology of artifacts. Lanzara and Morner (2005), for example, develop their argument also referring to the code itself and to the licenses based on copyleft. There is a growing literature discussing both how the inner configuration of the code affects FOSS projects' organizational features (e.g. Dalle et al., 2009; MacCormack et al., 2006), and how FOSS licenses influence social processes taking place within projects (Gambardella and Hall, 2006; Teli and De Paoli, 2006). In this paper I will focus only a subset of these artifacts: mailing list messages. I do this for the sake of clarity, as the process I will describe becomes more intuitive adopting this point of view. However, identical arguments —mutatis mutandis—can be done for all the other artifacts produced by the FOSS community. For example, in the case of pieces of code, their organization in modules reflects the interpretation of the technical procedures the creators had in mind, their beliefs on the properties of the product, and the social rules guiding their common action, i.e. the structures.

2.5. Inside "enactment": the crucial role of "negotiation of meanings"

The key feature behind structure propagation is the peripheral members' enactment of structure inscribed into threaded messages contained in a FOSS project's mailing list archive (i.e. into the artifacts). How does such enactment take place?

The virtual environment typical of FOSS communication is almost always mediated by computers, asynchronous and often stored in publicly accessible virtual spaces. The "ethos" of the community, the stories, the representations of the product, and the experiences of the developers are all stored online and organized into threads of discussions. This means that the discussions leading to the emergence of the core's structures are captured by the messages and organized into a *discourse* that inevitably embodies those structures. The discourse arising from the messages can then be considered the specific form that the inscription of core's structure takes when dealing with this specific kind of artifact.

In the open virtual environment where FOSS is produced, mailing list messages have wide visibility, and every idea and opinion can spread very quickly (Finholt and Sproull, 1990). Everyone, including those peripheral members who do not participate directly in the debate, has a wide and easy access to all the details of the discussion. As a consequence, when peripheral individuals access the archives where mailing list messages are stored, they face the inscription of the structures of the core in the messages therein. They are exposed to an organized set of opinions and ideas (i.e. the discourse) that may affect their own beliefs and principles.

They start to judge others' opinions, and choose among them those that they think could be more in line with their values, or even move into a transformation process where their identity is defined

around a new point of view. Consistent with Orlikowski's (2000) view, the simple action of using these artifacts, i.e. browsing and reading the messages, and comparing one's own beliefs with those expressed in the opinions inscribed in the messages, are the very actions we can identify as the *enactment* of the structures undertaken by the community members during their everyday practice. To see this, consider that opinions, threads of the discussions, and the organization of power emerging from the interlocutors' interaction, are all factors that have been forged according to the structures regulating the interaction of core developers; and at the same time they prevent (i.e. constrain) or support (i.e. enable) certain lines of reasoning that the peripheral member can decide to adopt (or discard). They are the carriers of the core's structures.

In order to define this process in detail, the concept of *negotiation of meanings* elaborated by Wenger (1998a, 1998b) in the context of community of practice is particularly useful.

Community of practice theory has been widely used, not only in management (e.g. Amin and Cohendet, 2004; Brown and Duguit, 1991), but also in the FOSS-related studies (e.g. Cohendet et al., 2001; Edwards, 2001; Lin, 2003a, 2003b, 2004a, 2004b). The main idea behind the concept is that social structures are based on the continuous coevolution of the individuals' activities and representations of the self and of the context they are immersed in. The nexus of ties constituting a community of practice is, in fact, a twofold space. On the one hand, the common space is populated by the everyday life of the community, where artifacts are produced, tasks are performed, and interaction among members takes place. On the other hand, in the same space—and together with the first activities—individuals construct their 'representations of the world'. This last term reflects the system of meanings, the semantics, through which reality is organized and filtered to be intelligible to each individual.

Each individual representation is not constructed in isolation, but also captures the stimuli coming from the interaction with the participants in the social space where the individual is immersed (the community). The result is that the meaning that each individual will finally give to his or her idiosyncratic experience is interwoven with others' experiences and representations. As a consequence, community members modify the sense of who they are and the principles guiding their actions, updating the former with the experience they have lived, and reconstructing the latter around the new features of the collective enterprise. The whole process can then be conceived of as a link connecting individual identities and social interaction through a precise mechanism: the *negotiation of meanings*.[11]

[11] To further investigate how this process could work in practice in a virtual environment see Levy (1984), Preece

It is easy to see that what has been described at the beginning of this section as structures enactment is simply the negotiations of meanings defined above: In this context, enactment of the core's structures by peripheral members *is* the negotiation of meanings triggered by the periphery's exposure to the core's discourse. This process takes the form of an internal and reflective (Hemetsberger and Reinhardt, 2006) "confrontation" between the peripheral individual representations and the structures inscribed and conveyed by the public discourse (refer to the upper part of Figure 2 to see how this can be represented graphically).

It is important to stress that "enact" does not mean that the users of the artifact comply with the rules and visions consistent with the structures (Orlikowski, 1992). Or, using words adapted to the empirical case of mailing lists: Reading online discussions does not mean accepting the constrains and resources defined therein. Users retain their agency, i.e. the capability of acting out autonomously, to a certain extent independent of the structures, or even modifying the structures in their usage of the artifact. When individuals use a technology, the structures inscribed therein materialize into their hands, brought to life again in the form of configuration of constraints and capabilities imposed on the individuals' practices by the artifact construction (Orlikowski, 1992). "Enactment" merely means that during the usage of the artifact (i.e. reading the threads stored in the archives and confronting the opinions therein), the users act in such a way that the structures inscribed in the artifacts affect their practices (in our case, they trigger a negotiation of meanings). Individuals enact the structures coming from the core *through their negotiation of meanings* aimed at filling the discrepancies between their own opinions and those brought to their attention by the discourse. They are not bounded by the enacted structures, rather they confront them with their own beliefs, visions, and idea of the artifacts themselves (Orlikowski, 1992, 2000). This "uncertainty" will be taken into account in the next section of the paper, when the process will be explained in greater detail.

3. A conceptual model: A step-by-step description of the structure propagation process

In the previous sections I provided an overview of the main mechanism behind structures propagation. In this section I will try provide a finer-grain perspective on the same mechanism, singling out its "nuts and bolts".

(2000), and Rheingold (2000). For a wider perspective see, for example, Golden-Biddle and Rao (1997) and Tuomi (2001).

3.1. Zooming in on the first step: how the discourse triggers structures enactment

By and large, in the FOS world, every individual belonging to the community can always access the very core of a past conversation that other individuals had about a specific topic. Public conversations are written and stored, and very often further discussion refers to previous posts. Hemetsberger and Reinhardt (2006) give a vivid description of the lively debate FOSS developers and participants are able to produce:

"Through speech acts such as explaining, evaluating, rejecting, correcting, insisting on an opinion or defending it, programmers engage in processes of collective reflection that potentially result in new knowledge-building. When the community engages in a process of conceptualization, first ideas and future goals are presented and comments are requested. After such initial messages, lively interactive conversation occurs, with comments supporting and further elaborating on the idea. Community members also present different perspectives towards the problem, or point out flaws or even errors in the presentation. These feedbacks and comments are again commented on, and initiate collective reflection processes. Thus the conversation revolves around the construction of the problem itself" (Hemetsberger and Reinhardt, 2006, p. 203).

In searching for information on a particular topic related to the FOSS world, it is common to enter directly these conversations going through the threads of messages archived in forums or mailing lists. The following quote, again from the study by Hemetsberger and Reinhardt (2006), makes this process clear:

"Our findings show that newcomers engage in exploring those archives in search of answers to their technical problems. They find the discussions in chronological order, which helps them to re-experience the lines of thoughts of the discussants. Quite often it is not the content of the discourse but the lines of arguments that provide the most valuable insights for learners." (Hemetsberger and Reinhardt, 2006, p. 208)

"Stored conversations" are then the basic material upon which the discourse on FOSS is realized. In debates and conversations, the participants' system of meanings and values "emerge to the surface" when community members are pushed to express an opinion on the discussed topic. In the FOSS case, an example of this "emersion" process is given by the case analyzed by Elliott and Scacchi (2003). The authors report a debate between conflicting views on the use of tools that are not free software. Two quotes from that debate can give the reader a clearer idea of the capability of the online conversation to capture the emotional level of the discussion and the difference in the systems of values of the discussants:

<CyrilB> neilt: you are compromising our freedom by using non-free software: we can't modify and/or redistribute the source vector file. […]

<neilt> otoh i see no reason to avoid non-free software either if this is really a freedom thing then we should be free to use whatever we want in which every participant tries expresses her or his opinions and elaborates on them to convince the other to act in a certain way" (Elliott and Scacchi, 2003, p. 26)

Exposed to such a conversation, observers can feel the challenge to form their own idea on the basis of their own preferences and on the other "material" conveyed to them by the debate.[12]

For example, a FOSS user entering the archives of the community communication just to find a needed piece of information will be inevitably exposed to the electronic communication of the core members, to their set of visions of the world or of the product, opinions and arguments. In other words, the user will be exposed to the core's discourse. Being a peripheral member, the user will probably face— with higher likelihood than other members—a debate on topics that he or she did not considered before, was not aware of, or had just an abstract idea of. When exposed to such a debate, the user will suddenly be in need of answering questions and acquiring positions about topics that he or she had never thought of. In other words, the discourse conveyed by the debate stored in the community mailing lists or forums interact with the user's systems of opinions, meanings and representations. The negotiation of meanings stemming form this interaction is precisely the enactment of the structure described above.

A first example can be retrieved from Kloos's (2006) empirical work on weblogs, wikis, and social bookmarking. The author interviewed students using a weblog to make sense of the main contents of a course they were following. Kloos reports quotes such as:

*"Yes, I have been able to form a clearer picture of the course. The blog has helped me in creating this picture." (*Kloos, 2006, p. 97).

Another quote from the same source makes clear that this picture can be challenged precisely through the discovery of unexpected opinions, as described above:

[12] This mechanism can be better understood considering a similar (even if more emotional) process described by Victor Hugo in *Les Misérables*: "We may remain more or less open-minded on the subject of the death penalty, indisposed to commit ourselves, so long as we have not seen a guillotine with our own eyes. But to do so is to be so shaken that we are obliged to take our stand for or against", *Les Misérables* (1862), Trans. by Norman Denny, Penguin Books, (1976), p. 32.

*"There were many posts that were not directly related to the course. A form of mind expanding, yes […] New things that were posted on the blog could shed new light on subjects, introducing connections you might not have been aware of before." (*Kloos, 2006, p. 103).

The discussion Linus Torvalds and Andy Tanenbaum had on the embryo of Linux, one of the fundamental parts of the GNU/Linux operating system that has become the most diffused FOSS operating system for desktop, is another good example of such a debate.[13] That discussion took place publicly and different people participated in it. But its importance went well beyond the circle of interacting developers. Today it is has become a fundamental piece of the FOSS history. That debate affected the vision of FOSS and of the community of thousands of individuals precisely through their experience of the FOSS discourse.

Also the argument between free software advocates (mainly gathered around the Free Software Foundation [FSF]) and open source software proponents (represented by the Open Source Initiative [OSI]), is another illustration of this process at work. Dahlander (2007) gives a precise description of the debate: "The FSF rests upon an ideology that strongly urges that information should be free…. FSF and OSI are united that the source code must be available, but disagree about many underlying beliefs. OSI felt that it opened the possibility for firms to commercialize and make money out of FOSS.… Pragmatic supporters of free software have been more willing to accommodate firms. Pragmatic arguments relate to the benefits of open code compared to closed code … rather than ideological motives that software ought to be free" (Dahlander, 2007, pp. 926–927). This debate concerns the essence of the community, and is considered fundamental by community members. In the FLOSS-EU survey (Ghosh et al., 2002), 48% state "I think of myself as a part of the Free Software community"; 32.6% say "I think of myself as a part of the Open Source community"; and only 19.4% of the sample does not care. The FLOSS-US survey (David et al., 2003) presents similar numbers: "I identify more with the Free Software community" has been marked by 31.4% of the responders, while 31.5% marked "I identify more with the Open Source Software community".[14]

The debate around this topic is then a crucial arena where different visions of the community and of the whole FOSS movement meet, contaminate, and confront one another. Yet, it is not likely that members at the periphery have thoroughly thought about this argument. As the analysis of the developers' motivations dynamics has shown (Glott, 2004; Glott et al., 2004; Rullani, 2006; Shah, 2006), most of the peripheral members are also users of the code, much less sensitive than core

[13] See http://www.oreilly.com/catalog/opensources/book/appa.html.
[14] To further explore the terms of the debate, see Giuri et al. (2002), Stallman (1998), Weber (2004) and the Web site of the *Open Source Initiative* (www.opensource.org/) and of the *Free Software Foundation* (http://www.fsf.org).

members to the ideological features of the FOSS movement. It is then not very likely that peripheral members are fully aware of such a distinction before facing the debate, where instead they will experience the expositions to messages relative to this unknown topic. The debate peripheral members are likely to face on this issue will be of the same level of "heat" as the one shown previously, and it will thus push them to form their own opinions and choose a position.

Notice that the urge to take a position under the stimulus of the debate is not a general processes valid for all possible circumstances. The possibility that the exposure to the debate will be powerful enough to trigger a negotiation of meanings in a peripheral member's experience of the FOSS community is directly connected to his or her "sensitivity", both to the topics stored in the online discussions, and to the media used for interaction, i.e. computers.

If the individual's vision of the world has no connection to the topic discussed in the community or is not open to anything different than, for example, finding a specific piece of information he or she needs to use the software, it is very unlikely that the dynamics described above can be realized. The role of interest in the common enterprise—at least in potential and in abstract terms—is an essential component of every community of practices (Wenger, 1998b). At the same time, if the individual is interested, the possibility for the negotiation of meanings to happen is positively associated to the degree of "sensitivity" the individual has with respect to CMC. If the individual is not familiar with this typology of communication or is not able to project the online discussion into a representation of the social environment that produced it, again the process described above is very unlikely to happen.

However, this does not mean that what has been said until now is the description of a marginal event. The reflexive process I described is an 'immanent' process. As the sociological and philosophical literature show, the processes upon which these forms of reflexivity are based are *innate*, and in this sense 'necessary' (Habermas, 1971, 1987; Wenger, 1998b; see also Fougère, 2004, for an account of this with respect to Bakhtin's theory). This means that they are present in every individual, even if their importance can differ across different environments and topics. The FOSS community is just one possible social space, and it may not be so relevant for a certain typology of individuals. Moreover, anthropological studies relative to CMC (e.g. Carbone and Ferri, 1999) have shown that CMC cannot be considered as something "less" than face-to-face communication. It is a different way of communicating, but is still able to convey strong feelings and emotions (Rheingold, 2000). However, each individual reacts differently to CMC. This results in a different perception of the quality and quantity of interaction, i.e. a different perception of the

"thickness" of the relationship. If an individual is not reactive to CMC, participation—and especially his or her peripheral participation—in the FOSS community is unlikely to be able to trigger reflexivity processes. On the contrary, more reactive individuals will be involved enough in the social environment of the community to trigger those processes.

In other words, the process described above can be strong enough to affect a substantial number of individuals, but at the same time add another dimension to the dissipative property of FOSS model of innovation: In order for active developers to emerge, the community must have mobilized and "burnt" an even larger amount of resources (i.e. individuals) just to be able to activate a much smaller amount of them (i.e. those "sensitive" to the topics discussed and to CMC).

3.2. Zooming in on the second step: structures enactment and dissonance

To understand the consequences of the "turbulence" the negotiation of meanings brings the peripheral member into, it is useful to recall the concept of *dissonance*. Dissonance can take different forms, as, for example, 'moral' or 'expressive' dissonance (Kuran, 1998). In general, it represents the mismatching between the individual's identity components, be they behaviors, preferences, moral values, opinions, or traits. As Kirman and Teschl (2006), building on a series of different theories (e.g. Akerlof and Kranton, 2000; Higgins, 1987; Livet, 2004, 2006), argue, dissonance results in a "psychological well-being loss". In other words, it can be defined as the *cost of incoherence*.[15]

When the individual exposition to the debate undertaken by the community triggers the negotiation of meanings related to the participants' opinions this pushes that individual to confront the structures underpinning the discourse. If the set of structures is consistent with the perception of the individual, there is no effect. However, if the two are in contradiction, the resulting friction has the effect of creating the individual's dissonance.

To place this in the context of FOSS, it is useful to briefly describe the case of Spip, an OSS project oriented to the production of Internet publishing software (Demazière et al., 2007a, 2007b). The project's core members have a strong political vision that is well expressed by the following quote from the project's charter:

[15] This also highlights the difference between the concept of dissonance based on internal mismatching on components that requires a solution in terms of internal consistency, and the idea of cognitive distance mentioned previously, that instead calls for the homogeneity of different individuals' conceptualizations of a problem with the purpose of sharing the same understanding. The concept of cognitive distance is more "interactive" than that of dissonance, and thus more useful in studies more focused on less peripheral members. For example, Muller (2006c) presents a simulation model in which FOSS community members are endowed with different behaviors and different levels of reputation. During the interaction, they aggregate around different behavioral rules to reduce cognitive distance, resulting in final outcomes that vary from complete convergence to convergence in subgroups that are not able to reconcile their views.

"*[…T]he participation to the Spip-zone must take place within the framework of the goals and values promoted by the initial Minirézo project, and notably to promote and defend freedom of speech for all on the Internet, to remain defiant towards financial interests, and to respect the identity of each and everyone. […] This site is not a development platform for military or business-oriented versions of Spip, which would change its nature. Neither is it designed to be used as a communication or advertising media for consultants*" (Demazière et al., 2007a, pp. 47–48).

This view, however, is in contrast with the fact that individuals related to Spip on the basis of their profession and business are an important component of the users and peripheral contributors of the project (Demazière et al., 2007b). This clash of views generates precisely the dissonance discussed above in that typology of peripheral members:

*"I am in love with the Spip community ...[b]ut I'm an outsider in the community because I've not yet been able to find the right words to explain that while I'm 100% for the participation in this project, I still need to earn a living (Armand)" (*Demazière et al., 2007b, p. 15).

As evidenced by the previous quote, peripheral members with a background different from that of the community suffer a "well-being loss" due to their inability to merge their own original identity with the community structures (Figure 2 tries to capture this mismatching graphically).

This quote also suggests another point. The interviewee seems to be actively searching for a way to make his beliefs align with the basic values expressed by the community core. This is because, in general, individuals experiencing dissonance can reduce it by changing the different elements of their identities to reestablish coherency (Kirman and Teschl, 2006). Also Kuran (1998) argues that one way of reducing dissonance is to align private preferences (what the individual *does* believe) to public preferences (what the individual *shows* to believe). In the present context, this process can be considered *internalization*, i.e. a change in one's personal values (and eventually possible actions) to conform to those underpinning the position the he or she has chosen. The result of this move is the experience of rule compliance, or alignment (Wenger, 1998b). Notice that this last passage does not require an actual action. The adoption of the external structures as rules of behavior provide the individual with a new set of behavioral patterns even when none of it is turned into action (as is often the case for peripheral members), but remains in the background as a procedure to be applied when needed. Through this experience, peripheral members absorb into their systems of values the structures propagating from the core of the community. Again, Figure 2 provides a graphical aid to the understanding of this passage.

3.3. Zooming in on the third step: moderators and obstacles to the process

3.3.1 Outcomes other than structure internalization

Of course this is just one possible outcomes of the process, as a first look at Figure 2 may have already shown. Even in the case of an individual "sensitive" to the debate conveyed by CMC, the negotiation of meanings mediating between the peripheral member's identity and the discourse can lead to an outcome different from internalization. If the gap generating dissonance is too wide, the tension can be ineffective in pushing the homogenization of the core's structures with the peripheral individual's identity. Kuran (1998) argues that a second mechanism able to reduce dissonance is *revolt*, defined as exposition of "knowledge and feelings that had tended to be concealed" (Kuran, 1998, p. 152). In this case, the individual can decrease his or her dissonance when entering the debate and exposing his or her opinion.

While Kuran's analysis is applied to nations and states where membership, together with its rights and obligations, is acquired at the moment of one's birth and difficult to cancel, in the present context the focus is on a community whose membership is easily manageable. Thus, in the FLOSS context, a third mechanism for dissonance reduction is also possible: a member can decrease his or her dissonance simply by *leaving* the community.[16]

When peripheral members experience a high level of dissonance, the probability that they move along an outward trajectory (Wenger, 1998b) and exit the community increases. The other outcome that acquires higher probability is the individual entering an inward trajectory (Wenger, 1998b), triggering or accelerating possible legitimate peripheral participation processes the individual could be involved in (Lave and Wenger, 1991). In this case, those who experience dissonance through the *debate* engage in the debate *directly* questioning the established discourse, which now becomes the center of a new, deeper debate. Elliott and Scacchi's (2003) example discussed above represents one of these cases.

The set of possible behaviors is, however, not limited to dissonance reduction mechanisms such as internalization, revolt, and leaving. Another possibility is that the level of dissonance may be high enough to prevent structures' internalization, but not high enough to result in active reactions such as those described. In this case, Kuran's (1989, 1995) conceptualization can help in understanding what could be the outcome of such a process. In his framework, the situation depicted above brings

[16] Notice that the three outcomes recall the framework adopted by Muller (2004, 2006c), i.e. Hirschman's concepts of loyalty, voice, and exit. In Muller's study, a developer's disagreement with the cognitive model adopted by the other community leaders results either in the developer's exit from the community, the voicing of the complaint, or a change of behavior on the push of loyalty toward the community. The same reasoning can be moved from the level of cognitive models to that of dissonance, and from core developers to peripheral members.

about an unresolved mismatch between the individual's private and public preferences, and this generates the accumulation of the resulting dissonance. The mismatch between identity components is simply left aside and not expressed in behavior (that conform to the publicly expressed preferences, both in potential and in act), but the cost of incoherence is not diminished. This means that, in our case, the peripheral individual does not significantly change behavior, but accumulates dissonance, probably still following an orbiting trajectory around the community (Wenger, 1998b).

This situation cannot last for long. Other episodes will make the accumulated dissonance emerge again. Some others may even increase it, expanding the discrepancy between the individual's established values and the structures the debate conveys.

As Kuran (1989, 1995) shows in his studies of revolutions, dissonance can remain latent in many individuals. In this case, a relatively insignificant trigger event, such as a small number of individuals questioning the status quo and showing to "inactive" individuals experiencing accumulated dissonance that disagreement is more widespread than expected, can result in what he calls a *revolution* (Kuran, 1989, 1995, 1998). The same process can be reframed in this context. When a certain number of peripheral members enter the debate and question the decisions of the core, other peripheral members who accumulated dissonance see that disagreement is not confined to their own individual opinion. This can act as a trigger event, and other peripheral members could follow and enter the debate as well. A new debate is now in place, this time questioning the very essence of the community[17] (see Figure 2 for a representation of all the outcomes).

3.3.2. Introducing Invisibility and Atomization

When internalization takes place, structure propagation is effective. When the outcome of the negotiation of meanings resembles one of the processes described above, instead, structure propagation is endangered. If the process leading to such outcomes could be managed, then the core could take action and correct an ineffective propagation. This could be possible only if the forces passing through the social body of the periphery were actually detectable. However, the periphery exhibits two properties, namely *atomization* and *invisibility*, that make it difficult for the core to infer the proportion of individuals in each one of the stages described above, and "hide" the periphery's path of evolution. These properties act as "veils" covering the movements of the periphery until a time when it would be very costly to move onto another track, thus endangering the process of structure propagation. (this is why both properties have been introduced in Figures 1 and 2 as moderators).

[17] Notice that this establishes a positive feedback that reinforces the frictions in the debate. This is in line with Kuran's

3.3.3. Invisibility

The FOSS community is indeed a virtual community. Contrary to the physical space, where mere presence is already a means of interaction and transmitting information relative to status or roles gives context to action, in the virtual space the act of observing others' behavior is usually not detected (Finholt and Sproull, 1990; Zhang and Storck, 2001). Members of a virtual space become visible only if they *act* in that virtual space. But peripheral members are, by definition, only minimally active, i.e. minimally visible. This perspective can be better defined using a theater as an example. Being the protagonist on stage, the core does not see the periphery in the audience. On the contrary, the periphery beholds the core and decides to applaud, to whistle, or to leave independently, to a certain extent, of the core's intentions.

3.3.4. Atomization

The periphery cannot be considered an organized group of individuals. Most of the time peripheral participants will make their decisions on an individual basis and relate them to what they observe in the core. In other words, individuals in the periphery will have an idiosyncratic answer to those activities of the core that they are exposed to. Some will leave while some will enter the community following a normal 'legitimate peripheral participation' process (Lave and Wenger, 1991) or a much more radical process questioning the community rules. Others will simply remain in the orbit around the core absorbing its structures. Peripheral individuals relate only to the core, and not to one another.

A similar perspective is put forward by West and Lakhani (2008) in their analysis of the concept of community. While investigating the role of interactions between community members, the authors ask if a user community à la von Hippel (1988) can still be considered a community when user-innovators provide their innovations to the focal firm but do not interact among themselves. The problem the authors consider relates to group dynamics: social bodies that are completely atomized, i.e. whose members do not relate to one another, and exhibit an aggregate behavior that is just the sum—and not the complex combination—of each component's idiosyncratic behavior. The lack of group dynamics makes the collective outcome a fragmented object, difficult to predict: It is not easy to have a vision of what the periphery is and what processes are moving it when signals do not conform to recognizable aggregate patterns and do not mirror aggregate properties of a collective. Signals are mixed and turn into "noise". The core, comprised of the actors on stage, can be considered a group working together. On the contrary, the periphery (the public) is mostly

---

(1989) description of the revolutionary process. The resulting difficulty in predicting the revolutions is also consistent with what is presented here.

composed of stand-alone individuals or small cliques who observe the play and decide to whistle, applaud, or leave on the basis of each individual's experience.

3.3.5. Atomization and invisibility as constraints to the propagation process

In terms of the dynamic interaction with the core, atomization and invisibility can lead to a situation in which the core may be unaware of the periphery's characteristics and its reception of the core's structures. Invisibility implies that the possible failure of the propagation process will not be easily detected. A decrease in the number of people orbiting around the project, for example, would coincide with the unnoticed disappearance of already invisible individuals. At the same time, the few signals the core is able to capture from the few feedbacks the periphery is able to broadcast are very likely not to be enough to infer what peripheral individuals' behaviors and intentions are as a whole. Atomization makes it very hard to relate some individuals' behaviors and expressions to those of the whole periphery, as any individual can be thought of as being representative only of her or his idiosyncratic situation. This amounts to a situation where the core is likely not to perceive the real size of most of the process at work in the periphery, and is often unable to receive feedback on the propagation of its structures. The lack of feedback implies that the core would not be able to detect the possible gaps opened between its activities and the periphery until it is too late.

An interesting case describing these points is offered by Alan Cox's (1998) discussion of the Linux 8086 project. Cox says:

"*The problem that started to arise was the arrival of a lot of (mostly well meaning) and dangerously half clued people with opinions - not code, opinions"* (Cox, 1998, online).

The instability of the situation was rooted in the disproportionate numbers of peripheral members who were unable to enter the core's activities and dynamics in a productive way. With the project's growth, the number of peripheral members increased beyond the core's capabilities to transfer its structures to the periphery, so that most of the peripheral members' contributions where incapable of matching the core's needs. The productive comments were overwhelmed by noise. In such a situation the answer of the core was to cut out the periphery, decreasing the noise but becoming deaf to any possible positive contribution:

"*…the real developers have many of the other list members in their kill files*[18] *so they can communicate via the list…. It ceased to be a bazaar model and turns into a core team"* (Cox, 1998, online).

As a further example one can recall what has been said about the Netscape's initially unsuccessful

[18] This technique prevents the developer from receiving any e-mails from specific addresses (those indicated in the "kill

opening of the code. The company had to go through a serious and "painful" period of failure in order to understand that the mismatching between the structures it was implementing and those perceived by most of the peripheral members was a serious issue, even after this point had been raised and voiced by many peripheral individuals.

4. Conclusions and discussion

The paper presented an analysis of the relationship between the core and the periphery in the FOSS innovation model. With respect to the existing literature, the novelty is that it tried to apply an opposite view, considering *the core* as the "*sparring partner*" of the periphery.

The paper defined at first the properties of the periphery in the specific context of FOSS (e.g. *dissipation*, David and Rullani, 2008; Lanzara and Morner, 2005), and then tried to uncover the different functions the periphery can perform thanks to those properties.

Structuration theory (Giddens, 1984)—as reframed by Crowston et al. (2005) in the context of FOSS—has been used to argue that those functions can be productive only when the same *structures* (i.e. procedures, values, and roles) applied in the core are correctly perceived by the peripheral members of the community. As peripheral members do no participate in the community activities, and their actions are characterized by low levels of interaction, I argued that when dealing with the periphery, structure propagation is possible as a result of *structure inscription into artifact*. To show how this is possible, I elaborated a conceptual model where *structure enactment* (Orlikowski, 2000), the process through which structures are "brought back to life" from the artifact, is redefined as a specific case of *negotiation of meanings* (Wenger, 1998b), where the peripheral members "confront" the structures emerging from the artifacts. To go more in-depth in the definition of the process leading to peripheral members' structures internalization, I integrated this theoretical background with the concept of *dissonance* (e.g. Kuran, 1998).

Finally, I have shown that the described process can have many different outcomes, not all of them leading to the realization of structure propagation. With respect to this, I analyzed what factors can endanger the propagation of the core's structures to the periphery, and found that two other properties of the periphery have such roles: *atomization* and *invisibility*.

The fact that the structure propagation process is "moderated" by the very properties of the periphery and by the inner mechanisms of the process itself opens the door to the discussion of possible strategies that managers of online communities can adopt to foster the division of labor between the core and the periphery.

---

file").

Even if in a partly different setting, the same problem of weak peripheral signals and of the creation of strategies able to capture them has been discussed by Brown (2004). In delineating the characteristics of a strategy able to recognize and retain the value produced by the periphery, the author stresses the importance of the construction of "**listening posts**", able to join the core and the periphery into "a social fabric that encourages disparate points of view and that affords creative abrasion between them" (Brown, 2004, p. 147).

Another strategy could be implementing procedures able to **filter the signals** coming from the periphery. Recall again the case of the Linux 8086 project: Cox argues that the total closure implemented by the core to diminish the noise weakened the project. Instead, "*Given a better ratio of active programmers to potentially useful wannabe programmers would have rapidly turned some of the noise into productivity*" (Cox, 1998, online).

A possible solution could have been adopting a "joining script" (von Krogh et al., 2003) able to single out useful contributions (and contributors) making them visible "above the noise". This has been Linus Torvarlds's strategy, common also to other projects (e.g. Freenet: von Krogh et al., 2003), and based on the idea that the provision of useful source code was the prerequisite to become relevant for (or "visible to") the core.

"In the Linux case […] as the project grew people who would have turned into 'The committee for the administration of the structural planning of the Linux kernel' instead got dropped in an environment where they were expected to deliver and where failure wasn't seen as a problem. To quote Linus [Torvalds, Linux's founder] 'show me the source'"[19] (Cox, 1998, online).

The overall lesson we can learn from this is that only when the core is able to create channels through which the periphery can be listened to in an efficient manner, the production of the latter can match the former's activities and transform the noise into a creative outcome.

Again referring to the example of the theater, we can recall a practice sometimes used in *metatheatrical* performances aimed at creating a peculiar bond between the public and the actors. For example, in some representations of the play *Tonight We Improvise* (Pirandello, 1930) in order to make the boundary between the actors and the public (the "fourth wall") fall, the former enter the audience and mix with the latter. The points of view of the public (the periphery) and that of the

[19] This is a process that works only for that particular part of the periphery that is able and willing to provide code to the project. As Cox (1998) posits: "Don't forget non programmers. … forgotten people who maintain web sites, change logs, mailing lists and documentation…. Linus says 'Show me the code'. That is a narrow view of a real project. When you hear 'I'd love to help but I can't program', you hear a documenter. When they say 'But English is not my first language' you have a documenter and translator for another language." This last passage is crucial. When the periphery is recognized as a much broader area, as done by Berdou (2007), it is brought on stage because it becomes clear how the periphery can be useful for a series of different activities. Having in mind only coding is a too narrow view of the FOSS

actors (the core) become interwoven, increasing the perception the actors have for the atmosphere inside the whole theater as well as intensifying the transmission to the public of the emotions and message of the play. This metaphor helps us to lead the discussion toward an interesting conclusion: caring about the periphery can be an important path toward success, as Pirandello's Nobel Prize suggests.

Figures

Figure 1. Properties, functions, conditions and moderators in the division of labor between the core and the periphery

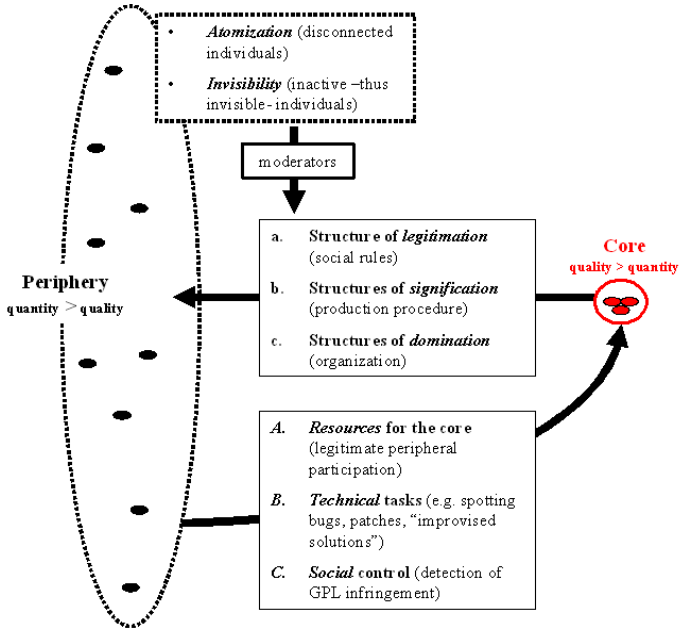
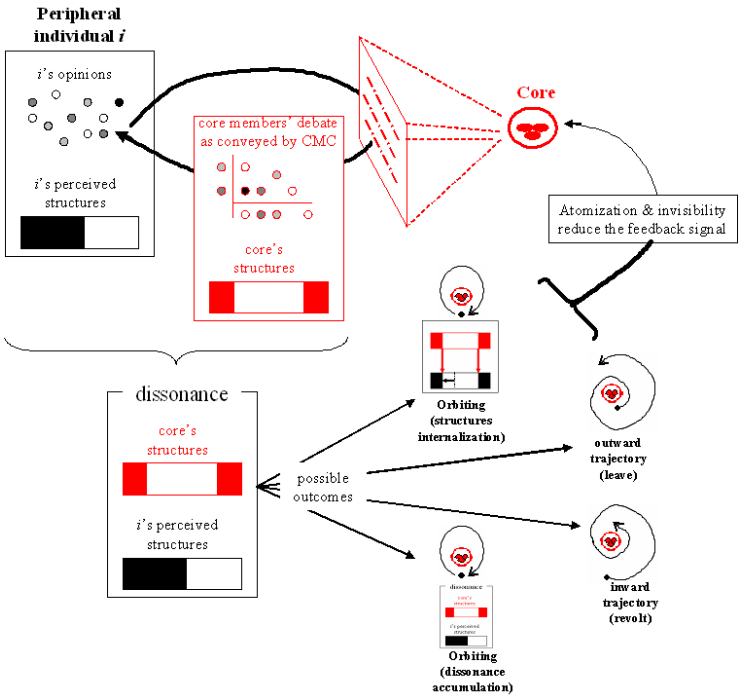


Figure 2. Dynamics of Structures propagation from the core to the periphery



model of innovation that neglects the many resources found at the periphery.

References

Akerlof, G.A., Kranton, R.E., 2000. Economics and identity. Quarterly Journal of Economics 115 (3), 715–753.

Amin, A., Cohendet, P, 2004. Architectures of Knowledge: Firms, Capabilities and Communities. Oxford University Press: Oxford, UK.

Augustin, L., 1999. Mozilla: fatal blow to open source. Linux Magazine 1 (2): 16–20.

Baldwin, C.Y., Clark, K.B., 2006. The architecture of participation: does code architecture mitigate free riding in the open source development model? Management Science 52, 1116–1127.

Berdou, E., 2007. managing the bazaar: commercialization and peripheral participation in mature, community-led F/OS software projects. PhD. Diss., London School of Economics and Political Science, Department of Media and Communications, London, UK.

Bessen, J., 2006. Open source software: Free provision of complex public goods. In: Bitzer, J., Schröder, P.J.H., (Eds.), The Economics of Open Source Software Development. Elsevier, Amsterdam, The Netherlands, pp. 57–81.

Bonaccorsi, A., Rossi, C., 2005. Licensing schemes in the production and distribution of open source software: An empirical investigation. Paper presented at DRUID Tenth Anniversary Summer Conference, 27–29 June 2005, Copenhagen, Denmark.

Brown, J.S., 2004. Minding and mining the periphery, Long Range Planning 37: 143–151.

Brown, J.S., Duguit, P., 1991. Organizational learning and community-of-practice: Toward a unified view of working, learning, and innovation. Organization Science 2, 40–57.

Carbone, P., Ferri, P. (Eds.), 1999. Le comunità virtuali, [Virtual communities]. Mimesis: Milan, Italy.

Cohendet, P., Creplet, F., Dupouët, O., 2001. Communities of practice and epistemic communities: a renewed approach of organizational learning within the firm. Paper presented at the Workshop on Economics and Heterogeneous Interacting Agents June 2001, Marseille, France.

Cox, A., 1998. Cathedrals, bazaars and the town council. Slashdot, at: http://features.slashdot.org/article.pl?sid=98/10/13/1423253 (accessed March 2009).

Cramton, C.D., 2001. The Mutual Knowledge Problem and Its Consequences for Dispersed Collaboration. Organization Science, 12 (3), 346-371.

Crowston, K., Heckman, R., Annabi, H., Masango, C., 2005. A structurational perspective on leadership in free/libre open source software teams. Proceedings of the 1st international conference on open source systems, Genoa, Italy, 9–15.

Crowston, K., Howison, J., 2005. The social structure of free and open source software development. First Monday, 10 (2).

Crowston, K., Wei, K., Li, Q., Howison, J., 2006. Core and periphery in free/libre and open source software team communications. Proceedings of the 39th Annual Hawaii International Conference on System Sciences, 6.

Dahlander, L., ,O'Mahony, S.C., 2008. Progressing to the center: The antecedents and consequences of lateral authority. Research paper 07-08, University of California at Davis Graduate School of Management.

Dahlander, L., 2007. Penguin in a new suit: A tale of how *de novo* entrants emerged to harness free and open source software communities. Industrial and Corporate Change 16 (5) 913–943.

Dahlander, L., Mckelvey, M., 2005. Who is not developing open source software? Non-users, users, and developers. Economics of Innovation and New Technology 14 (7), 617–635.

Dalle, J.M., David, P.A., Rullani F., 2009. Linking coordination, motivations and code structure in successful open source projects: A 'stigmergic' approach. To be presented at the Academy of Management, August 7 – 11, 2009, Chicago, IL, USA.

David, P.A, Rullani, F., 2008. Dynamics of innovation in an "open source" collaboration environment: Lurking, laboring and launching floss projects on Sourceforge. Industrial and Corporate Change 17 (4), 647–710.

David, P.A., Waterman, A., Arora,S., 2003. The free/libre/open source software survey for 2003. (Preliminary draft, September 2003). Stanford Institute for Economic Policy Research, Stanford University, CA., quoted with authors' permission. Available at: http://www.stanford.edu/group/floss-us/report/FLOSS-US-Report.pdf

de Laat, P.B., 2005. Copyright or copyleft? An analysis of property regimes for software development. Research Policy 34, 1511–1532.

Demazière, D., Horn, F., Zune, M., 2007a. The functioning of a free software community: Entanglement of three regulation modes – control, autonomous and distributed. *Science Studies* 20 (2), 34–54.

Demazière, D., Horn, F., Zune, M., 2007b. Individual and organizational approaches of voluntary participation in floss: An analysis based on the case of SPIP. Presented at the workshop The diffusion of FLOSS and the organisation of the software industry: From social networks to economic and legal models. May 31st – June 1st, Nice, France.

Edwards, K., 2001. Epistemic communities, situated learning and open source software development. Department of Manufacturing Engineering and Management, Technical University of Denmark, Lyngby, Denmark. Available at: http://opensource.mit.edu/papers/kasperedwards-ec.pdf

Elliott, M., Scacchi ,W., 2003. Free software developers as an occupational community: Resolving conflicts and fostering collaboration. Proceedings of the ACM International Conference Supporting

Group Work (Group'03), Sanibel Island, FL, 21–30.

Finholt, T., Sproull, L., 1990. Electronic groups at work. *Organization Science* 1, 41–64.

FLOSS-EU survey, at http://www.infonomics.nl/FLOSS/floss1/stats.html (accessed March 2009).

FLOSS-US survey, at http://www.stanford.edu/group/floss-us/ (accessed March 2009).

Fougère M., 2004. Organizing (with) thirdness: A dialogic understanding of bicultural interactions in organizations. Working paper 504, Swedish School of Economics and Business Administration, Helsinki, Finland.

Gambardella, A., Hall, B., 2006. Proprietary vs public domain licensing of software and research products. Research Policy 35(6), 875–892.

Ghosh, R.A., Glott, R., Krieger, B., Robles, G., 2002. Free/libre and open source software. Part IV: Survey of developers. International Institute of Infonomics, Berlecom Research GmbH. Available at: http://www.infonomics.nl/FOSS/report/Final4.pdf

Giddens, A., 1984. The Constitution of Society: Outline of the Theory of Structuration. Polity Press: Cambridge, UK.

Giuri, P., Ploner, M., Rullani, F., Torrisi, S., 2008a. Skills, division of labor and performance in collective inventions: Evidence from the open source software. Paper presented at the EARIE Conference, Porto, Portugal, 1–4 September, 2005.

Giuri, P., Rocchetti, G., Torrisi, S., 2002. Open source software: from open science to new marketing models an enquiry into the economics and management of open source software. Working paper, Laboratory of Economics and Management, Sant'Anna School of Advanced Studies (LEM), Pisa, Italy. Available at: http://www.lem.sssup.it/WPLem/files/2002-23.pdf

Giuri, P., Rullani, F., Torrisi, S. 2008b. Explaining leadership in open source software projects. Information Economics and Policy 20 (4), 305–315.

Glott, R., 2004. Towards integration of research approaches to FLOSS communities. Oxford Workshop on Libre Software (OWLS), Oxford Internet Institute, 25–26 June. Available at: <www.oii.ox.ac.uk/fiveowlsgohoot/postevent/owls-presentation_r.glott.pdf >

Glott, R., Ghosh, R.A., Krieger, B., 2004. Motivations of free/libre and open source developers. International Infonomics Institute Working paper, University of Maastricht, The Netherlands.

Golden-Biddle, K., Rao, H., 1997. Breaches in the boardroom: Organizational identity and conflicts of commitment in a nonprofit organization. Organization Science 8, 593–611.

Habermas, J., 1971. Knowledge and Human Interests. Beacon Press: Boston, MA.

Habermas, J., 1987. The Theory of Communicate Action: Lifeworld and System: A Critique of Functionalist Reason. Boston: Beacon Press: Boston, MA.

Hemetsberger, A., Reinhardt, C., 2006. Learning and knowledge-building in open-source

communities: A social-experiential approach. Management Learning 37 (2), 187–214.

Higgins, E.T., 1987. Self-discrepancy: a theory relating self and affect. Psychological Review 94 (3), 219–340.

Jensen, C., Scacchi, W., 2005. Modeling recruitment and role migration processes in OSSD Projects. Proceedings of the 6th International Workshop on Software Process Simulation and Modeling, St. Louis, MO, May 2005.

Kirman, A. Teschl, M., 2006. Searching for identity in the capability space. Journal of Economic Methodology 13(3), 299–325.

Kloos, M., 2006. Comm.unities.of.prac.tice 2.0:How blogs, wikis, and social bookmarking offer facilities that support learning in practice in communities of practice. Master Thesis, Program Business Information Systems, Universiteit van Amsterdam, The Netherlands.

Kock, N., 2004. The psychobiological model: towards a new theory of computer-mediated communication based on Darwinian evolution. Organization Science 15 (3), 327–348.

Kogut, B., Metiu, A., 2001. Open-source software development and distributed innovation. Oxford Review of Economic Policy 17 (2), 248–264.

Kuran, T., 1989. Sparks and prairie fires: A theory of unanticipated political revolution. Public Choice 61(1), 41–74.

Kuran, T., 1995. The inevitability of future revolutionary surprises. American Journal of Sociology 100 (May), 1528–1551.

Kuran, T., 1998. Social mechanisms of dissonance reduction. In: Hedström, P., Swedberg, R. (Eds.), Social Mechanisms: An Analytical Approach to Social Theory. Cambridge University Press, New York, pp. 147–171.

Lakhani, K.R., 2006. The Core and Periphery in Distributed and Self-organizing Systems. PhD Diss., Sloan School of Management, MIT, Cambridge, MA.

Lakhani, K.R., Jeppesen, L.B., Lohse, P.A., Panetta. J.A., 2006. The value of openness in scientific problem solving. Working Paper, 07-050, Harvard Business School, Boston, MA.

Lakhani, K.R., Panetta, J.A., 2007. The principles of distributed innovation. Research publication No. 2007-7, Harvard Law School, Cambridge, MA.

Lakhani, K.R., von Hippel, E., 2008. “Microdistributed” innovation projects. Presented at the DRUID/Scancor Conference on Distributed Innovation, Stanford, CA, 27–28 March.

Langlois, R.N. Garzarelli, G., 2008. Of hackers and hairdressers: Modularity and the organizational economics of open-source collaboration. Industry & Innovation 15 (2), 125–143.

Lanzara, G.F., Morner, M., 2005. Artifacts rule! How organizing happens in opens source software projects In: Czarniawska, B., Hernes, T. (Eds.), Actor-Network Theory and Organizing,

Copenhagen Business School Press, Copenhagen, Denmark, pp. 67–90.

Lave, J., Wenger, E., 1991. Situated Learning. Legitimate Peripheral Participation. Cambridge University Press, Cambridge, UK.

Lazear, E.P., 1999. Globalization and the market for team-mates. *Economic Journal* 109 (454),15–40.

Lee, G.K., Cole, R.E., 2003. From a firm-based to a community-based model of knowledge creation: The case of the Linux Kernel development. *Organization Science*, 14(6), 633–649.

Levy, S., 1984. Hackers. Bantam Doubleday Dell: New York.

Lin, Y., 2003a. Culture, Creativity, and Innovation: practices of locally-crafted software. Presented at the joint seminar SATSU and HCI group at computer science and psychology departments on 20 May, University of York, UK. Available at http://www.york.lug.org.uk/talks/yuwei_satsu/satsu_paper.html (accessed May 5th, 2005).

Lin, Y., 2003b. The institutionalisation of hacking practices. Ubiquity 4 (4), 2. Available at: http://www.acm.org/ubiquity/views/y_lin_1.html

Lin, Y., 2004a. Contextualising knowledge–making in Linux user groups. First Monday 9 (11). Available at: http://firstmonday.org/issues/issue9_11/lin/index.html

Lin, Y**.,** 2004b. Epistemologically multiple actor-centered systems: or, EMACS at work! Ubiquity 5 (1). Available at: http://www.acm.org/ubiquity/views/v5i1_lin.html

Livet, P., 2004. La pluralité cohérente des notions de l'identiteé personnelle. Revue de Philosophie Economique 9 (1), 29–57.

Livet, P., 2006. Identities, capabilities and revisions. Journal of Economic Methodology 13(3), 327–348.

MacCormack, A., Rusnak, J., Baldwin, C.Y., 2006. Exploring the structure of complex software designs: An empirical study of open source and proprietary code. Management Science 52, (7), 1015–1030.

MacIntosh, R., MacLean, D., 1999. Conditioned emergence: A dissipative structures approach to transformation. Strategic Management Journal 20 (4), 297–316.

Mateos-Garcia, J., Steinmueller, W.E., 2003. The Open Source Way of Working: A New Paradigm for the Division of Labour in Software Development? Open Source Movement Research working paper 1 (January), Science and Technology Policy Research, University of Sussex, Brighton, UK. Available at: http://www.sussex.ac.uk/spru/publications/imprint/sewps/sewp92/sewp92.pdf

Mockus, A., Fielding, R.T., Herbsleb, J., 2000. A case study of open source software development: The Apache server. Proceedings of the 22nd International Conference on Software

Engineering, 263–272.

Muller, P., 2004. Autorité et gouvernance des communautés intensives en connaissances : une application au développement du logiciel libre. Revue d'Economie Industrielle, 106. (2nd Trimester), 49–68.

Muller, P., 2006a. How does leadership support the activity of communities of practice? BETA - Document de travail 2006–14, April.

Muller, P., 2006b. Reputation, leadership and communities of practice: The case of open source software development. In: Dolfsma, W., Soete, L. (Eds.), Understanding the Dynamics of a Knowledge Economy. Edgar Elgar, Cheltenham, UK, pp. 77–101.

Muller, P., 2006c. Reputation, trust and the dynamics of leadership in communities of practice. Journal of Management & Governance 10, 381–400.

Ngamkajornwiwat, K., Zhang, D., Koru, A.G., Zhou , L., Nolker, R., 2008. An exploratory study on the evolution of OSS developer communities. Proceedings of the 41st Annual Hawaii International Conference on System Sciences, 7-10 January, Waikoloa, HI, 305.

O'Mahony, S., 2003. Guarding the commons. Research Policy 32 (7), 1179–1198.

O'Mahony, S., Ferraro, F., 2006. Managing the boundary of an open project. In: Powell, W., Padgett, J. (Eds.), Market Emergence and Transformation. MIT Press: Cambridge. MA.

O'Mahony, S., Ferraro, F., 2007. The emergence of governance in an open source community. Academy of Management Journal 50 (5), 1079–1106.

Orlikowski, W.J., 1992. The duality of technology: Rethinking the concept of technology in organizations. Organization Science 3 (3), 398–427.

Orlikowski, W.J., 2000. Using technology and constituting structures: A practice lens for studying technology in organizations. *Organization Science* 11 (4), 404–428.

Pirandello, L., 1930. Questa sera si recita a soggetto. Mondadori, Milano, Italy. [Trans. by Campbell, J.D., Sbrocchi, L.G., 1987. Tonight We Improvise; and Leonara Addio! Canadian Society for Italian Studies: Ottawa, Canada].

Preece, J., 2000. Online Communities. Wiley, New York.

Prigogine, I., Stengers, I., 1984. Order Out of Chaos; Man's New Dialogue with Nature. Bantam Books, New York.

Raymond, E., 1998a. The cathedral and the bazaar. First Monday 3 (3). Available at: http://www.firstmonday.org/issues/issue3_3/raymond/index.html

Raymond, E., 1998b. Homesteading the noosphere. First Monday 3 (10). Available at: http://www.firstmonday.org/issues/issue3_10/raymond/#d12

Rheingold H., 2000. The Virtual Community. MIT Press, Cambridge, MA.

Rullani, F., 2006. Dragging developers towards the core. How the Free/Libre/Open Source Software community enhances developers' contribution. Presented at the CCC Thirteenth Annual Colloquium for Doctoral Student Research, 19–21 May, Lausanne, Switzerland.

Shah, S., 2006. Motivation, governance and the viability of hybrid forms in open source software development. Management Science 52 (7), 1000-1014.

Singh, P.V., Fan, M., Tan, Y., 2007. An empirical investigation of code contribution, communication participation and release strategies in open source software development: A conditional hazard model approach. FLOSS working paper. Available at: http://flosspapers.org/327/

Stallman, R., 1998. Why 'free software' is better than 'open source'. Free Software Foundation. Available at: http://www.gnu.org/philosophy/free-software-for-freedom.html

Teli, M., De Paoli, S., 2006. Boundaries and diversity in free and open source software, presented at The Second International Conference on Open Source Systems, June.

Tuomi, I., 2001. Internet, innovation, and open source: Actors in the network. First Monday 6 (1). Available at: http://firstmonday.org/issues/issue6_1/tuomi/index.html

von Hippel, E., 1988. The Sources of Innovation. Oxford University Press, New York.

von Krogh, G., Spaeth, S., Lakhani, K.R., 2003. Community, joining, and specialization in open source software innovation: a case study. Research Policy 32, 1217-1241.

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

Wenger, E., 1998a. Community of practice: Learning as a social system. Systems Thinker 9 (5), 1–10.

Wenger, E., 1998b. Community of Practice: Learning Meaning and Identity. Cambridge University Press, Cambridge, UK.

West, J., Lakhani, K.R., 2008. Getting clear about communities in open innovation. Industry & Innovation 15 (12), 223–231.

Zhang, W., Storck, J., 2001. Peripheral members in online communities. Presented at the AMCIS Conference, Boston, MA.

Zirpoli, F., Rullani, F., Becker, M.C., 2009. Coordinating distributed innovation processes: The case of the automotive and open source software industries. To be presented at the DRUID conference, 17 - 19 June 2009, Copenhagen, Denmark.