Managing the Bazaar: Commercialization and peripheral participation in mature, community-led Free/Open source software projects

Evangelia Berdou

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Media and Communications

London School of Economics and Political Science

June 2007

I, Evangelia Berdou, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

....

Signature

Abstract

The thesis investigates two fundamental dynamics of participation and collaboration in mature, community-led Free/Open Source (F/OS) software projects - commercialization and peripheral participation. The aim of the thesis is to examine whether the power relations that underlie the F/OS model of development are indicative of a new form of power relations supported by ICTs.

Theoretically, the thesis is located within the Communities of Practice (CoP) literature and it draws upon Michel Foucault's ideas about the historical and relational character of power. It also mobilizes, to a lesser extent, Erving Goffman's notion of 'face-work'. This framework supports a methodology that questions the rationality of how F/OS is organized and examines the relations between employed coders and volunteers, experienced and inexperienced coders, and programmers and non-programmers. The thesis examines discursive and structural dimensions of collaboration and employs quantitative and qualitative methods. Structural characteristics are considered in the light of arguments about embeddedness.

The thesis contributes insights into how the gift economy is embedded in the exchange economy and the role of peripheral contributors. The analysis indicates that community-integrated paid developers have a key role in project development, maintaining the infrastructure aspects of the code base. The analysis suggests that programming and non-programming contributors are distinct in their make-up, priorities and rhythms of participation, and that learning plays an important role in controlling access. The results show that volunteers are important drivers of peripheral activities, such as translation and documentation. The term 'autonomous peripherality' is used to capture the unique characteristics of these activities. These findings support the argument that centrality and peripherality are associated with the division of labour, which, in turn, is associated with employment relations and frameworks of institutional support.

The thesis shows how the tensions produced by commercialization and peripheral participation are interwoven with values of meritocracy, ritual and strategic enactment of the idea of community as well as with tools and techniques developed to address the emergence of a set of problems specific to management and governance. These are characterized as 'technologies of communities'. It is argued that the emerging topology of F/OS participation, seen as a 'relational meshwork', is indicative of a redefinition of the relationship between sociality and economic production within mature, community-led F/OS projects.

Acknowledgements

The PhD often appears as a solitary undertaking. One thing that PhD students, however, learn very quickly, is that impossible to maintain the degree of focus and dedication required for its completion without the help of many people.

My supervisor, Professor Robin Mansell, provided her unfailing support and guidance, while challenging me to move beyond my intellectual comfort zones. I would not have been able to do this without her.

I am indebted to the Greek State Scholarships Foundation which funded the first three years of my study. I would also like to thank the DBE and OPAALS European projects, and especially Dr. Paolo Dini, that supported my research over the last two years.

I am grateful for the time and openness of all the F/OS contributors who participated in my research. I would like to especially thank Thomas Zander, Federico Mena Quintero, Scott Wheeler, Telsa Gwynne, Luis Villa, Christian Schaller, Gerard Briscoe, Yiota Tsatsou and Ellen Helsper, for their generous assistance in various aspects of my study.

Dr. Gordana Uzelac and Elizabeth Van Couvering were there for me through the best, and the worst, at late nights and early mornings.

Ute, Marianna and Constantina believed in me before I did.

Lastly, I am forever grateful to my husband, Alexandros, and my parents for their love and support.

London, June 2007

Table of contents

List of Figures

List of Tables

1 Introduction

1.1 Overview

This chapter provides an introduction to the research, which examines power relations in Free/Open source (F/OS) software projects. The F/OS model of development is often considered to be one of most promising new models of production and labour organization to emerge as a result of the widespread availability of information and communication technologies (ICTs) in the last 15 years. Part of its significance lies in that it may be applicable to other domains of production, such as content production, as indicated by the success of Wikipedia. However, existing explanations of its success tend to adopt a limited perspective that fails to address important issues of access, control and power.

The central question addressed in this study concerns the structure and dynamics underlying the F/OS model of development which is often perceived as a socio-technical process. F/OS software is developed by online communities of globally distributed contributors, the majority of whom are volunteers who rarely meet face to face. Despite this, they develop strong collaborative ties and abide by the principles of cooperation. F/OS developers are often said to adhere to the ideal of meritocracy, be driven by the principle of reciprocity and be guided by peer-review practices. This shared value system is often considered to underlie F/OS's emergent social organization, where the most skilful and dedicated contributors achieve the highest status among their peers. If adopted unquestioningly this perspective can lead to the idea that F/OS projects are self-contained and reasonably homogenous and that they adhere to their own functional logic. However, it is argued in this thesis that this perspective provides an incomplete picture of F/OS development because it fails to account for a number of issues and tensions associated with cooperation and participation, such as those concerned with different levels of access or with the differential value that may be attributed to different types of contribution.

To address this weakness in the literature, the present research seeks to answer:

- What are the dynamics that underlie F/OS software projects?
- How are power relations between various actors constituted and maintained at different levels of interaction?

1.2 *The F/OS model of development: brief history and major characteristics*

The term 'open source' describes:

a) Software protected under special copyright licences aimed at ensuring availability and free (re)distribution of the source code. Source code refers to the set of instructions written by developers that make up a program. Proprietary software is distributed only in the form of object code, the machine readable translation of the source code, which is required for computers to run programs. Open source software is distributed both as source and object code.

b) A process of software development that incorporates some unique technical and social characteristics, such as the ability of users to suggest new features, report faults in programs, etc.

c) A movement based on the ideals of the hacker culture which is premised upon the freedom to use, create, and tinker with software, and the values of the gift culture, such as the ideal of reciprocity (Kollock, 1999).

Prominent examples of open source software include the GNU/Linux operating system, the Apache server program and the Python computer language.[1] Initially, most of the software produced by the F/OS movement had an infrastructural character. As Castells (2001) indicates this meant that its users consisted of programmers and system administrators and very few applications were addressed to the average, non-technical user. However, this is rapidly changing. F/OS is being adopted by a growing number of public and corporate organizations, and reaching a wider and more diverse non-technical user base compared to its earlier phases of development.[2]

Open source is frequently used as an alternative to, or in conjunction with the term 'free software' (as in FLOSS - Free Libre Open Source Software), as a reference to the movement and the special characteristics of its development process. However, these two terms can also be used to describe two different agendas in the F/OS movement. Free software (http://www.gnu.org/philosophy/free-sw.html, last accessed 05/05/07), which originates from Richard Stallman and the Free Software Foundation, is often associated with an emphasis on the ethical and philosophical commitments of the movement and, specifically on the notion of 'freedom'. The term open source

[1] The GNU/Linux (see, http://www.linux.org/, last accessed 05/06/06) operating system is the F/OS software equivalent of Windows. The Apache webserver is a F/OS program that allows computers to handle HTTP requests from clients (http://httpd.apache.org/, last accessed 05/06/2006). Python is a high-level computer programming language created by Guido van Rossum in 1991 (see http://www.python.org, last accessed 03/04/2006).

[2] Some prominent examples include the LinEx project developed in Extremadura, Spain which developed its own Linux distribution used in school computers across the region, and the development of corporate targeted shrink wrap Linux distributions developed by Novell Inc. and by other F/OS distributors.

(http://www.opensource.org/docs/definition.php, last accessed 05/05/07), which was originally coined by Bruce Perens and Eric Raymond, is employed to depict what they and others consider to be a more commercially viable approach that is less restrictive from a business perspective, with regard to licensing terms, than FLOSS.

F/OS is protected under copyright that restricts the way the software is used, modified and distributed. F/OS licences were developed in order to prevent anyone hijacking the code. There are two main types of F/OS licences: copyleft and non-copyleft (Lessig, 2002). The best known example of a copyleft licence is the GNU Public License (GPL). The viral nature of the GPL, that is the requirement to licence the modified software under the same terms as copyleft, has been regarded as a disincentive to commercial exploitation. Non-copyleft licences, a subset of which conforms to the Open Source Definition (http://www.opensource.org/docs/definition.php, last accessed 05/05/07) were created to address the desire for easier commercial exploitation. These licences protect the right to freely distribute and modify code, but impose fewer restrictions on subsequent use and distribution (Lessig, 2002).

Its recent popularity belies the long history of the F/OS movement. The tradition of sharing and cooperation that underlies the F/OS model of software production dates back to the early days of software development in the 1950s. It is founded on an engineering culture that grew out of major academic institutions such as MIT, the University of California at Berkeley and the Carnegie Mellon Institute. This tradition is associated with the set of values that form the basis of what is widely known as the 'hacker culture'. Among the prominent features of this culture are the pursuit of technical excellence and the joy of creativity (Castells, 2001; Himanen, 2001; Levy, 1984). In agreement with Richard Stallman, Castells (2001: 46-47) notes that: "paramount in this set of values is freedom. Freedom to create, freedom to appropriate whatever knowledge is available, and freedom to redistribute this knowledge under any form and channel chosen by the hacker".

The apparent undermining of this freedom by commercial software companies through their withholding of the source of software programs, deprived hackers of the opportunity to tamper with it. This motivated Richard Stallman to build his Free Software Foundation. One of the Foundation's initial goals was to create a functioning clone of UNIX, the operating system that was developed in the 1960s and 1970s by a group of employees at AT&T (American Telephone and Telegraph Company), which included Ken Thompson, Dennis Ritchie and Douglas McIlroy, that was commercialized in the 1980s.

The development which marked the take-off of the F/OS movement was introduced by a Finnish computer student called Linus Torvalds. Torvalds created the kernel of the Linux operating system, a program that has been regarded as the leading potential challenger to Microsoft Windows. Torvalds helped introduce a decentralized collaborative model for software development founded on the principle of reciprocity. Linux's rapid pace of development called into question, the validity of doctrines of traditional software development and management, such as 'Brooks' law' (Brooks, 1995). According to this law the complexity and communication costs of a software project rise with the square of the number of developers, but the rate of productivity follows a linear function.

In their book *Understanding Open Source Development* Joseph Feller and Brian Fitzgerald (2002: 84) provide a useful description of the main characteristics of F/OS. F/OS development is said to be:

- "Parallel, rather than linear;
- involves large communities of globally distributed developers;
- utilizes truly independent peer preview;
- provides prompt feedback to user and developer contributions;
- includes the participation of highly talented, highly motivated developers;
- includes increased levels of user involvement;
- makes use of extremely rapid release schedules."

The term parallel refers to a process of development whereby the code can branch out into multiple lines of development and to the process through which the same code development may be pursued by different people simultaneously. Parallel development can apply to such activities as adding new features, reporting and fixing bugs[3] or testing, and to different levels of the code. An advantage of this method is that it allows developers to alternate between stable and experimental code releases and simultaneously to pursue different solutions to a problem or desired program feature. It is assumed that it is the best available technical solution that is incorporated in the code. It has been suggested that the ability to revert to older and more stable versions of the code encourages experimentation and innovation. The replication of effort associated with parallel development is regarded as evidence of an abundance of labour, provided by the principle of reciprocity underlying the gift economy (Barbrook, 1998; Raymond, 2000a).

As indicated in the next section, some authors have argued that these characteristics are not unique to F/OS. Proprietary software development can share many of the practices of the F/OS model, including the globally distributed character of development.

1.3 F/OS: changing perceptions, unanswered questions

This section provides an account of the insights emerging from the F/OS literature by focusing on the theoretical, methodological and empirical contributions

[3] Bugs are faults in a program.

that are related to the issues of organization, power and control. The breadth of perspectives is considered in section 1.3.1, which focuses not only on the success of the movement, but also on expectations in relation to its potential benefits. Section 1.3.2 provides a synthesis of the findings relating to the issues of power, coordination and control in F/OS development.

1.3.1 Reception and appropriation of the F/OS phenomenon

F/OS is an evolving phenomenon. This is reflected in the multitude of perspectives from which it has been examined. F/OS has become emblematic of processes that are widely associated with the 'network economy' (Shapiro and Varian, 1999). F/OS appears to some to provide evidence of many of the ways in which the promises inherent in discourses concerning the 'new economy' (Herzenberg, 1998) or the 'information society' (Webster, 1995) are being realized.

To be more precise, F/OS has been regarded as:

- A revolutionary ***method of software development*** that could potentially provide the solution, be the 'silver bullet', to the problem of software productivity which has led to the software crisis (Bauer and Pizka, 2003; Bezroukov, 1999a; Feller and Fitzgerald, 2002; Raymond, 2001a, b; Weber, 2004).

- A platform for a ***novel innovation process***, one that is distributed and user-driven, which appears to bridge between the consumers and producers of information (Garcia and Steinmueller, 2003b; von Hippel, 2002; von Hippel and von Krogh, 2003; Weber, 2004).

- A movement that has led to the ***emergence of a new commercial actor***, community-driven open source projects, which indicates an evolution in the relationship between community, culture and commerce. (O'Mahony, 2003, 2002; Weber, 2004).

- A generator of ***new business models*** that have the potential to change the structure of the software industry (Behlendorf, 1999; Krishmamurti, 2005).

- An ***engine of development*** and empowerment of the public sector (Dravis, 2003; Schmidt and Schnitzer, 2002; Weerawarana and Weeratunga, 2004).

- An ***alternative to the rationality of the exchange economy, the gift economy*** which expresses the essence of community, promotes the idea of software as a public good and is based on the principle of anticipated reciprocity and peer-to-peer exchange (Berguist and Ljungberg, 2001; Kollock, 1999; Newmarch, 2001; Raymond, 2000a; Raymond, 2001a, b).

- A ***paradigm for online, collaborative, distributed*** work, often referred to as the 'bazaar' model of development, that is characteristic of the network society and which is potentially transferable to other domains of knowledge and areas of production (Bezroukov, 1999b; Dafermos, 2001; Feller and Fitzgerald, 2002; Garcia and

Steinmueller, 2003b; Ljundgberg, 2000; Moon and Sproull, 2002; Stalder and Hirsh, 2002).

These claims will be discussed in greater detail in the succeeding chapters of this thesis. It is only necessary here to note that this synthesis shows that initial assumptions about the F/OS model, such as its pure meritocratic or even collective basis, have been challenged. Nevertheless, there is significant scope for a critical examination of aspects of F/OS development that have been understudied, both in relation to the internal dynamics of cooperation and to the links of F/OS development with other actors.

1.3.2 An examination of the organizational, collaborative aspects of F/OS

The complexity of the F/OS phenomenon is reflected in the richness of the academic literature. Nevertheless, there are some persistent characteristics and weaknesses in the theoretical, empirical and methodological orientation of this literature towards the issues of coordination and control.

Most examinations of the F/OS phenomenon rely on a combination of three main perspectives:

- *Cultural, ideological perspectives.* These draw on influential practitioner accounts (Raymond, 2001b), emphasizing the role of ideology and the desire to participate in the gift economy as the primary factors in the success of the movement (Elliot and Scacchi, 2003b; Hemetsberger and Reindhart, 2004; Stewart and Gosain, 2006).
- *Socio-technical perspectives.* F/OS projects are regarded as systems with technical and social aspects that involve a recursive process of technology and social shaping. These tend to emphasize the infrastructural aspects and technical characteristics of F/OS developments as the prerequisites for the ability to scale up F/OS projects and reach a high level of complexity (Baldwin and Clark, 2005; Ducheneaut, 2005; Shaikh and Cornford, 2003, 2004).
- *Techno-economic perspectives.* Few F/OS studies employ formal economic models and several studies are framed by insights from organizational studies, labour economics and the economics of innovation (Dalle and David, 2005; Dalle, et al., 2005; Garcia and Steinmueller, 2003a, b; von Hippel, 2002; von Hippel and von Krogh, 2003).

Those writing within the cultural perspective tend to argue that the success of the movement is founded on the values of the hacker culture, which emphasize access and freedom of information, reciprocity and meritocracy. With a few exceptions, such as the study conducted by Magnus Berguist and Jan Ljundberg (2001) which views the F/OS movement as an 'attention economy', whose members ratchet up their reputation in order to attract participants to their projects, most argue for the primacy and centrality of hacker beliefs in creating a framework of trust and collaboration (Elliot and Scacchi, 2003b; Hemetsberger and Reindhart, 2004; Stewart and Gosain, 2006). This view is problematic in that it offers a cyclical explanation of why F/OS works, but does not address important questions related to issues of control and power. This issue is addressed in more detail in Chapter 2. Cultural perspectives also dominate explanations of the way new developers become incorporated into communities. The process of integration is consequently often described as a process of socialization into the established practices and values of the hacker culture (von Krogh, et al., 2003). However, the barriers to access in F/OS arguably not only involve an appropriation of relevant values, but may also express the dynamics of power that are shaped by the development process itself.

Those writing from a socio-technical perspective argue that coordination and consensus are achieved mostly through the technical design of F/OS. For example, Baldwin and Clark (2004) suggest that the modular architecture of F/OS mitigates free-riding, reduces the costs of communication, provides rewards, and encourages repeated interactions. Although Weber (2000) agrees with Baldwin and Clark that it is the architecture of the technical system that underlies F/OS organization, Reagle (2004) turns this argument on its head suggesting that bug tracking tools embody the values of how a community should come to agreement or even disagreement. Similarly, Shaikh and Cornford (2003; Shaikh and Cornford, 2004) have examined issues around version tracking tools reflecting governance and decision making in the Linux Kernel project community, suggesting that these tools create a learning environment. This work is often informed by a social construction of technology framework,[4] suggesting that the socio-technical interdependencies of F/OS source projects deserve greater attention. These authors provide useful insights into community development, but the work is limited in that it does not provide insights into the dynamics that comprise this recursive process of technology in the F/OS context.

Techno-economic perspectives include influential contributions, such as Lerner and Tirole's studies (Lerner and Tirole, 2002a, 2005) of the economics of F/OS and von Hippel's (2005) conceptualization of F/OS projects as horizontal, user-driven, innovation networks. In addition to investigating questions of motivation, coordination and conflict resolution, which are also studied within cultural and socio-technical perspectives, those writing within techno-economic frameworks tend to examine questions of resource allocation more systematically (Dalle, et al., 2005; Garcia and Steinmueller, 2003b). They also consider issues of F/OS software competitiveness in relation to proprietary software and whether F/OS projects meet the long-term needs of users in society. In most of these studies individuals are treated as the unit of analysis, especially where the focus is on the motivations of F/OS developers.

The question of 'why hackers do what they do' (Lakhani and Wolf, 2005; Roberts, et al., 2006b; Shah, 2003), namely why programmers contribute to

[4] See (Bijker, et al., 1987) Also (Bijker and Law, c1992).

developing software without monetary reward, pervades most research on the F/OS phenomenon. Many studies have suggested the prevalence of intrinsic rather than extrinsic motivations. Intrinsic motivations have been associated with activities that satisfy basic psychological needs, such as enjoyment, competence and control. In the case of F/OS developers intrinsic motivations are said to include the joy derived from problem-solving and learning, and the satisfaction that derives from contributing to a public good (Lakhani and Wolf, 2005).In contrast, extrinsic motivations are said to be needs, such as peer-recognition and material and monetary rewards. These claims have been challenged by empirical studies revealing that developers do obtain financial benefits from their involvement in projects (David, et al., 2003; Hann, et al., 2004)[5]. For example, Hann et al. (2004) indicate that the recognition earned through participation in the Apache project is associated with significantly increased wages. The results of some later studies suggest that F/OS development is driven by a mix of intrinsic and extrinsic motivations and that their balance can be influenced by a variety of technical and institutional factors, such as the licensing scheme (Aigrain, 2002; Lerner and Tirole, 2002b), and the personality of project leaders.

Most writings suggest that the success of F/OS projects hinges on the balance between intrinsic and extrinsic motivations and the private and the public (von Hippel and von Krogh, 2003). Researchers have begun to pay more attention to the implications of the corporate and public sectors in the funding of F/OS development (Schmidt and Schnitzer, 2002). Despite progress being made on elaborating a framework for understanding motivations underlying F/OS development, however, the focus on motivations and, specifically, on the dichotomy between intrinsic and extrinsic motivations, provides a rather narrow and instrumental view of the dynamics that influence cooperation in F/OS projects.

The methodology that predominates in F/OS research is quantitative. This is partly because of the availability of publicly archived data in mailing lists, code submissions and bug databases. Large-scale surveys of development communities (David, et al., 2003; Ghosh, et al., 2002; Lakhani and Wolf, 2002) are being complemented by studies using data from code repositories and Concurrent Version Systems (CVS)[6]. Their structure resembles a tree, providing programmers with the advantage of retrieving older versions of the programs that they are working on and the possibility of simultaneously exploring and working on different technical solutions. CVS is also used by non-programming groups within projects, such as translators and documenters who use them as a means of coordinating their work. However, not all aspects of F/OS development are captured by the CVS.

Research based on data mining from CVS and other public data sources falls into two main categories. The first consists of software engineering studies on performance related aspects of F/OS development, for example, code reuse, patterns

[5] This is examined in detail in Chapter 4, which focuses on the dynamics of commercialization.

[6] A CVS (Concurrent Versions System) is an open source version control software tool that allows developers to keep track of changes made to the code and to coordinate their work.

of source commits, and quality control (Neamtiu, et al., 2005; Sliweski, et al., 2005). The second consists of studies that draw on software engineering and social science and that focus on the structure and organization of F/OS communities. Although some of these utilize CVS data (Huang and Liu, 2005; Koch and Schneider, 2002; Lopez-Fernandez, et al., 2004; Spaeth, 2005), others use data from mailing lists (Crowston and Howison, 2005; Roberts, et al., 2006a). The majority of these studies rely on one source of information and assume that the projects' organization is based on internal dynamics of collaboration and participation. Moreover, social network analysis studies employing data from mail archives infer patterns of collaboration from patterns of communication. These studies have some limitations if the aim is to examine questions related to community organization and evolution.

Despite these methodological limitations some of these writings offer insights that challenge established ideas about F/OS development. For example, in contrast to the view that F/OS development is a highly collective endeavour, Ghosh and David's (2003) study on the Linux kernel indicates that development takes place in small groups. The idea that developers work in small groups is also supported by Milchmayr and Hill's (2003) study: 'Quality and Reliance of Individuals in Open Source Projects'. Whereas they embrace Raymond's (2001a: 19) view that "given enough eyeballs all bugs are shallow" they do not find that the development of software benefits from parallel development in the same way that debugging does. These findings are consistent with Koch and Schneider's (2002) study on effort and cooperation which demonstrates that individual developers work more or less in isolation, on different modules. The collective nature of cooperation in F/OS is also contested by Krishnamurti (2002) who concluded that many F/OS programs listed in SourceForge net[7] are produced by a handful of developers. Also, Howison et al. (2006) found that the centralization of projects, as defined by developers' communication patterns, varies widely, and that the team of individuals with the strongest communication ties rarely changes.

Studies using social network analysis frequently suggest that the social organization of F/OS communities is such that hierarchies and a core group of developers become established on the basis of talent and competence. An emergent view of F/OS takes up the biology metaphor in referring to self-organizing systems (Lakhani, 2006). These studies do not investigate the origins of these divisions and whether they are associated with structural factors, such as developers' employment status. They also generally fail to address the theoretical implications of the decentralized model of F/OS development. An exception is Demil and Lecoque's (2006) study which compares the F/OS bazaar with the governance mechanisms of firms, markets and networks. Demil and Lecoque argue that the bazaar governance structure, characterized by a specific contractual framework and by innovative coordination mechanisms, is potentially as efficient as other forms of governance.

[7] The SourceForge net is an online development and download repository for F/OS projects (see, http://sourceforge.net/, last accessed 05/05/2006).

These studies have been complemented by works that draw attention to the institutional and structural factors affecting meritocratic participation. O'Mahony and Ferraros's (2003) study of membership in the Debian[8] community indicated that changes to the social structure of the project affected the evolution of membership mechanisms and the determination of gatekeepers. In this project membership was institutionalized and criteria were established that defined that an individual was recognized as a member of the community on being given a CVS and an email account. This was deemed necessary in order to control the flow of new developers and to assure the quality of contributions. This study suggests that the introduction of new coders does not depend only on their ability to assimilate the values of the hacker culture, but also hinges upon structural factors.

In contrast, Daniel Stewart's (2004) study on patterns of peer certification in Advogato[9] indicates how important it is for members that want to achieve high status to work quickly to establish a positive reputation if they do not want to run the risk of being cast in a low status position. Stewart's findings undermine the idea of an individual proactively managing his/her own career, slowly progressing from a low to high status position. Instead, Stewart maintains that participants quickly decide about newcomers that immediately begin to work to establish a position in the community's social structure. However, Advogato is not an F/OS project; it is a community resource for free software developers and a research testbed for work on group trust metrics. Many newcomers are well-known developers with established reputations. Despite the study's methodological shortcomings, however, Stewart's findings are inconsistent with the 'small world hypothesis' that emerges from other studies (Koch and Schneider, 2002).

It is not only the collective and purely meritocratic basis of F/OS that is questioned in the literature. Garcia and Steinmueller (2003b) argue that many F/OS practices, also take place in proprietary contexts, such as in the case of commercially Distributed Software Development. This argument is supported by the observations of Divitini et al. (2003) and Fuggetta (2003), who suggest that F/OS projects share socio-technical similarities with proprietary software development, for example, forging alliances, organizational manoeuvres, and daily software builds.

With regard to authority in F/OS, the assumption underlying most studies is that it is dependent upon technical rationality and excellence, which is in line with the meritocratic view of collaboration. Garcia and Steinmueller (2003b), however, suggest that technical decisions often involve a degree of arbitrariness, although there are indications of the development of sets of heuristics and rules of thumb. They provide

[8] Debian (see http://www.debian.org, last accessed 05/05/2006) is one of the most popular community-led GNU/Linux distributions. The distribution bundles together different programs (operating system, drivers, Graphical User Interface, user applications) and provides an easy-to follow installation process that allows non-technical users to install and run a fully functional GNU-Linux system on their computers.

[9] Advogato is a site that serves as a community resource for free software developers. Its peer certification project is described by the founder and maintainer of the site as a "test bed for work on group trust metrics" (see http://www.advogato.org, last accessed 08/09/04.

a functional explanation for the management of complexity in F/OS projects, by drawing attention to their underlying hierarchies. The development of hierarchy, as in the emergence of an inner circle of developers is said to be a response to the growth of projects, both technically and socially. Garcia and Steinmueller's argument about models of distributed authority is indicative of the efforts to draw out the wider organizational implications of the F/OS model of development. In examining the F/OS model of development most studies view F/OS projects as contained, static systems, underpinned by their own functional logic.

One of the first studies to provide findings regarding F/OS-corporate interactions was conducted by Siobhan O'Mahony (2002). It indicated how the community-managed projects and companies "created a set of social arrangements that were new to both of them. Competing processes of mutual accommodation and preservation helped define the attributes of the new form: private non-profit software foundations" (O'Mahony, 2002: vii).

Very little of the work on how communities and companies interact, has focused on the question of how the increasing professionalization of the community, primarily expressed through the involvement of paid contributors, affects the dynamics of cooperation and participation. Most studies, do not take into account this potentially important structural aspect of participation.[10] As projects consolidate their relations with industry these issues will become increasingly important and F/OS source projects can no longer be conceived as closed systems that relate to the outside world only through specially designed interfaces.

In addition, most studies so far have focused on coders. However, as F/OS communities scale up, they need to mobilize other skills than programming. Documentation for developers, for example, is a crucial part of the effort to lower the barriers to participation for new volunteers. Translations of documentation and of program interfaces play a crucial part in program's dissemination. In addition, as F/OS reaches a wider, non-technical, user base, questions of usability and design are becoming more crucial. The issue of how relations between these different types of contributors are organized, therefore, will become increasingly important.

This section has highlighted insights from the literature with regard to issues of organization and control, which paints a rich picture of F/OS development. The highly collective and purely meritocratic basis of F/OS development and the innovativeness of its technical and organizational characteristics have been challenged. Many contributions draw attention to a core/periphery model of F/OS development in which a core group of programmers is responsible, substantiating arguments concerning the emergent nature of F/OS organization.

Not only are there significant pieces missing from the puzzle of what constitutes F/OS development, but the pieces seem to belong to other different

[10] David et al.'s (2003) preliminary findings from the FLOSS US survey, indicate e.g. that 20% of respondents earned money, either by developing full-time (14.6%) or part-time (6.1%).

puzzles. Research findings provide an incoherent picture. This may be due to the persistence of a functional perspective on the F/OS model of development in which F/OS developers are understood to operate in a space outside the commercial market. This gap in the literature comes about as a result of the narrow focus on experienced code contributors and by the view of new coder integration as a process of enculturation.

1.4 Significance of the research

Dominant discourses concerning the information society stress the increasing importance of ICTs in contemporary societies (IV European Union, 2006; United Nations ICT Task Force, 2004, 2005). The terms network information and knowledge societies are so frequently used and abused that the concepts have become almost invisible, empty of content and meaning. Emphasis is often placed on the quantitative aspects of the information societies that have been envisaged in recent decades or on how intensively technology pervades our everyday lives. However, these approaches give little insight into the substantive and qualitative transformations that are associated with today's information society (Webster, 1995).

To address this it is useful to draw attention to two phenomena that bring into focus two interrelated, qualitative transformations signified by the emergence of the information society. The first is associated with the emergence of the bazaar model of development, which, as we have seen, represents a new, decentralized model of production and distribution. This term was coined by Eric Raymond (2000) to describe the decentralized, bottom-up, almost organic, process of software production represented by the F/OS model of development The second relates to the generalization of the gift economy through the Internet (Leyshon, 2003).

Raymond (2000b) first used the term 'bazaar' in his famous essay 'The Cathedral and the Bazaar' in which he contrasted the F/OS model with the traditional model of software development which had been conceptualized by an influential software engineer, Frederick Brooks (1995) as a process resembling the building of a cathedral. Whereas in the cathedral model order is achieved through centralized control and adherence to a master plan, in the bazaar model, order seems to emerge organically through the complex interactions of a multiplicity of actors.

The gift economy refers to socially embedded forms of economic relations that are usually associated with pre-capitalist societies (Mauss, 1954; Polanyi, 1944) that have survived in modern societies (Cheal, 1988; Offer, 1997). Whereas the exchange economy consists of commodity and monetary exchanges taking place within a framework shaped by scarcity and the impersonal efficiency of markets, the gift economy is characterized by redundancy and constitutes a system of transactions between interdependent individuals on the basis of the principle of reciprocity. In addition, while the exchange economy is driven by the pursuit of individual gain, in the gift economy individuals give unconditionally in order to freely benefit from the collective effort. In F/OS the value system underlying this particular form of gift giving is often referred to as the 'gift culture'. In line with Cheal's argument that rather than being extinguished the gift economy has undergone a continuous transformation, many have argued that the networking potential of new technologies combined with the underlying values of high-tech groups have induced a generalization of the gift economy through the Internet (Barbrook, 1998; Leyshon, 2003; Veale, 2005). Benkler (2006) in fact talks about the F/OS model of development and of other forms of peer-to-peer production as a new form of 'social production'.

An instrumental definition of the notion of community is adopted in the present study. As is discussed in greater detail in Chapter 2, the idea of community is defined on the basis of the pursuit of a shared enterprise, that usually takes place within the workplace. This definition of community has its roots in organizational theories of learning and innovation (Wenger and Lave, 1991).

In this thesis it is suggested that these two phenomena, the bazaar organization and the extended boundaries of the gift economy, constitute two important aspects of the emerging information society which converge in the growing significance of the role of volunteer communities in knowledge production and innovation.[11]

Although these two phenomena provide an intuitive basis for the understanding of possible transformations in the processes of production and innovation, they do not help us to understand how this process is carried out through the F/OS community members' practices, or how to develop a framework for examining the multiple interconnections developing between different spheres of economic activity. The growing significance of F/OS software and the increasing importance attributed to peer-to-peer models of production and distribution underlain by the values of the gift culture, testify to the need for theoretical frameworks that will allow a more comprehensive understanding of how these communities structure participation and organize practice, and the relation of these emerging organizations with established organizations at the level of everyday development.

1.5 Primary case studies and methodology

This chapter has introduced the main characteristics of the F/OS phenomenon. The significance of F/OS for understanding the transformations associated with the rise of the information or network society were discussed and the main empirical and theoretical approaches in the literature identified together with gaps in our understanding of the dynamics of F/OS. The aim of this study is to investigate:

[11] This question is pursued in more detail in Chapter 2.

- What are the dynamics that underlie F/OS software projects?
- How are power relations between various actors constituted and maintained at different levels of interaction?

This chapter has highlighted trends in the examination of F/OS and arguments emerging from many of the existing studies. Given the persistent gaps in the understanding of the way power operates in F/OS, the research questions will be examined by focusing on the specific dynamics of participation and collaboration. The first is the impact of *commercialization*. The second is the issue of *access* and the *collaborative* ties ***between new and senior developers and between different types of contributors***.

The research examines these issues at the level of mature, community-led F/OS projects. Specifically, the primary case studies for this research are the GNOME (GNU Network Object Model Environment) and KDE (K Desktop Environment) projects. The aims of these projects together with their degree of technical maturity and the types of relations examined, frame the generalizability of the research findings.

GNOME and KDE are aimed at developing a complete graphical user interface (GUI) for the Unix operating system (in all its different variations), providing a host of end-user applications, for example, office suites, and developing the tools necessary to further develop both the platforms and the end-user applications. The parallel examination of two very similar projects challenges the dominant view that the organization of projects is dictated by their development needs.

In order to fulfil their aims GNOME and KDE have mobilized varied contributor communities, which include both programmers, and contributors with non-technical skill sets. The technical and the non-technical contributions that underlies development in KDE and GNOME are more balanced than in other types of F/OS projects, such as those that are addressed to a more specialized audience, mainly programmers and system administrators. It is reasonable to expect that the dynamics of collaboration between different types of contributors depend on the projects' primary audiences and the domain knowledge associated with the areas of their application.

Both these projects have reached a significant level of technical maturity, as indicated by their many releases and their long-standing presence. This means that their ties with the commercial world have had more time to develop than those of immature projects that have yet to prove their technical viability. Furthermore, the fact that they are community-initiated and led, differentiates them from corporately initiated F/OS projects that seek to invite voluntary contributions and which usually maintain a stronger degree of control over what is accepted in the code. These two characteristics have a considerable influence on the dynamics of cooperation between paid and voluntary contributors.

The research examines discursive and structural aspects of participation and cooperation by: a) tracing the interconnections of bodies of knowledge that give meaning to the experience of participants; and b) by mapping the practices and material factors that structure access and participation, including contributors' employment status.

To achieve this the research adopts an iterative, three-step research design that combines quantitative and qualitative methods. Phase 1 consists of 23 interviews conducted with a sample of four groups of participants: experienced and novice programmers, volunteer and paid developers, non-programmers and community members involved at a high administrative level. These interviews examined how contributors occupying different strategic positions in the community reflected on a similar set of issues, and elicited information on different aspects of development and cooperation. Phase 2 consists of three nested case studies, which involved 16 additional interviews, which examined in more depth aspects of commercialization and relations between experienced coders and new programmers, and programmers and non-programmers, which emerged through the interviews conducted in phase 1. In addition to the interviews, in phases 1 and 2 seven F/OS community events were observed.

Phase 3 consists of a quantitative analysis of patterns of contribution of GNOME Foundation and KDE e.V. [12] members and of the two project maintainer networks. This phase of the research aimed to examine specific hypotheses regarding patterns of volunteer and paid developer contributions and non-programmer and programmer contributions as they emerged through phases 1 and 2. One hypothesis was that paid developers usually focus on critical aspects of development.

1.6 Thesis Outline

The thesis is structured as follows:

In Chapter 2 the theoretical and conceptual framework of the research is presented and discussed.

In Chapter 3 the methodology of the research is outlined and the primary case studies and research design are presented in detail.

In Chapter 4 the analysis of the empirical findings relating to ***commercialization*** is presented.

In Chapter 5 the analysis of the empirical findings concerning ***access and cooperative relations between new and experienced programmers and between programmers and non-programmers*** is presented.

[12] The GNOME Foundation (http://foundation.gnome.org/, last accessed 01/05/07) is an organizational body within the GNOME project that coordinates releases and mediates between companies and the GNOME community. KDE e.V. performs a similar function for the KDE project. For more information on the two bodies see chapter 3, section 3.7.2.1.

In Chapter 6 the results of the quantitative investigation of the network of GNOME Foundation and KDE e.V. members and of the two projects' maintainer networks are outlined.

Chapter 7 provides a synthesis and a discussion of the empirical findings in light of the theoretical framework adopted and the principal research questions.

Chapter 8 discusses the contribution of the research in the context of related work, the limitations of the study and avenues for future research.

1.7 Conclusion

Overall, the research is designed to contribute to a better understanding of the dependencies developing between the gift and exchange economies and the dynamics of participation and cooperation that develop among different types of contributors, namely paid and volunteer developers, programmers and non-programmers and experienced and new programmers. A typology of engagement of paid contributors based on employment arrangements and community ties is developed and the new types of interfaces that develop between communities and corporate actors are highlighted. The research indicates that community integrated paid developers have a key role in project development and that, in addition, programming and non-programming groups are distinctive in their make-up, priorities and rhythms of participation, with volunteer programmers driving peripheral activities. Furthermore, the findings indicate that learning plays an important role in controlling access to the community's centre.

The analysis in this thesis shows how the discursive and structural tensions produced by commercialization, the integration of new coders and the involvement of non-programmers, is interwoven with specific programmes of action, tools and techniques, which are termed 'technologies of communities', to produce a complex topology of participation that creates different opportunities for control and agency. These technologies of communities, which are essential in establishing communities as objects and subjects of governance, are: the programme of meritocracy, the ritual and strategic enactment of the idea of community and the tools and techniques developed to address the emergence of a set of problems specific to management and governance.

2 Theoretical and conceptual framework

2.1 *Chapter overview*

This chapter sets out the theoretical and conceptual framework of the research that underpins the examination of the power relations in F/OS communities and more specifically those associated with the dynamics of commercialization and peripheral participation. This framework combines elements from three theoretical perspectives: communities of practice (CoP), Michel Foucault's conception of power and his methodology for studying power relations, and Erving Goffman's idea of face-work.

The chapter is structured as follows. Section 2.2 outlines the CoP perspective, indicating how it has been appropriated in the context of F/OS and specifies which elements of this perspective are adopted within the context of this research. The CoP perspective is critically assessed and further developed in order to apply it to the study of power relations between new and experienced members of a community, and relations between members of different communities. Section 2.3 focuses on Foucault's ideas on power, and particularly his ideas about the historical, contingent and relational character of power. It examines his methodology for studying power relations. In order to situate the discussion on power in F/OS, Foucault's conception of power is compared with Bourdieu's notion of habitus and Giddens' structuration theory (section 2.3.3). Section 2.4 focuses on Goffman's idea about team and individual face-work. The compatibility between the theoretical elements borrowed from Foucault and Goffman is discussed (section 2.4.3). Section 2.5 introduces the concept of meshwork, a concept that is used heuristically and analytically. The chapter concludes with an outline in section 2.6 of the conceptual framework within which the empirical part of the study is situated.

2.2 *The Communities of Practice perspective*

The primary framework for the research is drawn from the CoP perspective, an approach that was developed to account for forms of learning that take place outside formal education contexts. This theory has frequently been adopted to explain some aspects of F/OS projects because it provides an intuitive way of understanding the social organization of F/OS that is consistent with practitioners' explanations. Section 2.2.1 provides an overview of how the theory has been appropriated and highlights the research programmes within which the approach has been registered.

Section 2.2.2 focuses on the way the CoP approach is mobilized within the context of this research. In contrast to its more common culturally oriented appropriations, this research applies the CoP perspective to challenge the idea of F/OS projects as homogenous communities built exclusively around the practice of programming in which learning functions as an enculturation process. This is achieved through an examination of F/OS communities as constellations of practice and by highlighting learning as a structuring factor of participation.

Section 2.2.3 recapitulates how this application of the CoP approach provides a basis for developing a better understanding of the dynamics of cooperation in mature F/OS projects and the potential to augment this approach with respect to understanding the power relations between community members.

2.2.1 The CoP perspective in context

Since 1992 there has been an increasing interest in the role of community in knowledge production and innovation. Notions such as epistemic cultures, epistemic communities and CoP have been formulated in order to express what has been perceived as the increasing importance of communities as repositories of knowledge and cradles of innovation (Knorr-Cetina, 1999; Lesser, et al., 2000). The role of ICTs in this surge of interest has been instrumental, primarily because they allow some of the traditional knowledge communities to transcend the confines of space and time thus enhancing their reach and effectiveness, and because they enable the formation, or indeed, the design, of new communities.

It is perhaps worth noting that all these different notions, which in one way or another stress the link between community, work and creativity, have emerged in slightly different contexts. Peter Hass (1992) formulated the notion of epistemic communities within the framework of international policy coordination and International Relations (IR). Epistemic communities are networks of professionals with recognized expertise who share common normative and causal beliefs, common notions of validity and policy goals. Knorr-Cetina (1999) studied the role of epistemic cultures, "those amalgams of arrangements and mechanisms-bonded through affinity, necessity and historical coincidence-which in a given field, make up how we know what we know" (1999:1) primarily within the context of science and science institutions. In exploring the culture and practices of personnel maintaining Xerox copiers, Julian Orr (1996) coined the term 'occupational communities'. Finally, Wenger and Lave (1991) developed the idea of CoP as a theory of learning and socialization within and across traditional organizations.

Despite their slightly different points of origin, these four conceptual variations on the idea of occupational or epistemic community, share some important characteristics. All, for example, regard knowledge, culture and practice as being closely interrelated. Within this context, knowledge is culture, culture is practice, and knowledge is practice. All four notions referred to above are also underscored by the idea that reality, the reality of the community and the reality of the area in which it is activated, is socially constructed. The meaning of community is negotiated and co-constructed by its members and the community itself becomes the prime context within which mutual engagement works out common sense practices. These concepts are also based on the idea that a community's collective knowledge is greater than the sum of the knowledge of its individual members. And, in addition, these ideas are usually associated with a bottom-up view of how work and cooperative relations operate in institutions and organizations, which cuts across traditional boundaries and managerial structures.

The emphasis on the communal and cultural aspects of practice, prevalent in the CoP perspective, usually overshadows other aspects of community life and activity, namely issues of access and power within and across its boundaries. This is partly due to the strong administrative agenda that underlies most existing studies that adopt this approach, which regard CoP as vehicles for innovation and competitiveness. Wenger's (2002) book entitled *Cultivating Communities of Practice: a guide to managing knowledge* is indicative of this growing trend. As John Brown and Paul Duguid (2001a: 203) point out:

> The ideas of communities of practice has been taken up with a remarkable amount of enthusiasm. It nonetheless (or perhaps for this very reason) needs handling with care, much like the notion of 'culture' over which we hesitated above. On the one hand, much on the enthusiasm turns on the appeal of the word community, which Williams (1976) suggests can be a deceptive but "warmly persuasive word" (It is worth contemplating how wide the notion would have spread had Lave and Wenger decided to talk about a *cadre or* a *commune* of practice.)

Jacky Swan et al.'s (2002) study provides an alternative, more critical perspective of how managers attempted to construct a new community of practice as a vehicle for innovation by using the notion of communities as a rhetorical device for overcoming some of the competing interests of different professional groups. Their case study chronicled how the managers of a bioscience company tried to enrol different groups of medical professionals in the advancement of a new prostate cancer therapy. Their study highlights the shift that occurred in managers' practices when faced with powerful professions and limited organizational support; they "employed a strategy centered on constructing a community that was focused on the disease rather than on the product. Adopting such community building "reflected managers' lack of power to intensify innovation by other means" (Jacky et al., 2002: 477). This study is intriguing because it raises some important questions regarding not only the role and function of CoP, but also the hidden dynamics that invocation of the idea of community might imply.

2.2.2 *CoP and F/OS: opening up practice and structuring participation*

In this section we consider the strengths of the CoP perspective as the point of departure for an examination of power relations in F/OS projects. After highlighting the weaknesses of the theory and the way that it has been applied in the study of F/OS, attention is drawn to some of the elements that are adopted in this research, namely the conceptualization of F/OS projects as 'constellations of practice' and the reworking of the concept of the periphery.

The CoP approach offers an intuitive way for understanding the organization of F/OS projects. Its structural elements, for instance, the centre and periphery division, and its view of integration as socialization are compatible with a cultural view of F/OS, which argues that the success of the movement relies on the values of the hacker culture. Moreover, this view is consistent with practitioners' explanations regarding a bottom up view of power as in the bazaar metaphor, and the meritocratic basis of participation, which establishes a social hierarchy founded on the practice of programming. As a consequence, the CoP approach has been adopted in a number of F/OS studies, either as the principal frame of analysis (Elliot and Scacchi, 2003a; Samer and Wasko McLure, 2002; Sharma, et al., 2002) or as part of a wider theoretical agenda (Berguist and Ljungberg, 2001; Madanmohan and Navelkar, 2002).

With the exception of Berguist and Ljungberg's (2001) article 'The power of gifts: organizing social relationships in open source communities', which draws attention to the differential power dynamic of gift-giving and the unequal relationships that it establishes, most of these studies stress the communal and ideological elements of participation and consider learning as a process of enculturation. This approach often leads to rather circular explanations about how the F/OS model works. For example, in their article on resolving conflict and fostering collaboration in F/OS source projects, Elliot and Scaatchi (2003: 1) see conflicts arising as a result of different norms regarding the use of non-open source tools in the development and, at the same time, regard disputes as being settled through daily discussions which "serve to build and perpetuate the global community of GNUe[13] contributors as well as F/OSS developers in general".

The under-theorized aspects of power relations in approaches that adopt the CoP perspective in the study of F/OS reflect the poorly developed framework in the CoP literature for understanding power. In ***Situated learning and Peripheral Participation,*** the book that introduced the notion of 'communities of practice', Lave and Wenger (1991: 42) acknowledge that:

> The term 'communities of practice' is left largely as an intuitive notion, which serves a purpose here but which requires a more rigorous treatment. In particular unequal relations of power must be included more systematically in our analysis.

Given the original focus of the theory this is not surprising, but Lave and Wenger's programmatic statement has not been followed, and the issue of power has not been rigorously addressed within the CoP literature (Contu and Willmott, 2003; Fox, 2000). For example, Wenger's (1998) more systematic account of the CoP

[13] GNU Enterprise (GNUe) is a sub-project of the GNU Project. GNUe's goal is to create free "enterprise-class data-aware applications, (enterprise resource planners etc.) http://www.gnuenterprise.org/, last accessed, 22/03/07. GNU is an operating system composed entirely of free software. It was created by Richard Stallman (see chapter 1, section 1.2) and it is a recursive acronym that stands for GNU is Not Unix. Many GNU components are used by Linux.

perspective, published seven years the after his original contribution with Lave, concentrates on the social constructionist aspects of CoP and brings the issue of power to the fore primarily in relation to the process of identity formation.

The CoP perspective, therefore, was initially developed in order to account for and to systematize forms of learning that take place outside formal education, primarily within multiple work contexts. Strictly speaking, the idea of a community of practice implies a group of people involved in a shared practice. Lave and Wenger illustrate their ideas by five examples: Mayan Midwives in Yucatec, a group of non-drinking alcoholics, meat cutters in a supermarket, quartermasters in the US navy, and Vai and Golan tailors in West Africa. They suggest that these groups are based on triadic relations between 'masters' ("old timers"), 'young masters' or ("journeymen") and 'apprentices' (or "newcomers"), the dynamics of which are very different from traditional teacher-student relationships. Newcomers must essentially learn from the masters, but their learning involves some contribution to the community, usually in the form of execution of routine tasks. As the skills of the newcomers evolve and they adopt the ways of the community, they gradually move from the periphery, where if accepted, they hold the status of legitimate peripheral members, to the centre.

One aspect of power that is addressed by the CoP perspective, at least in Wenger and Lave's original contribution, is that of *access*. They examine this aspect of CoP through the example of the apprenticeship of meat cutters. In this case newcomers were granted the status of apprentices, but they were denied access to the more mature practices of the community. Thus newcomers were granted peripherality, but were denied legitimacy. Wenger also addresses the issue of participation by pointing to the difference between peripherality and marginality. Peripherality is conceived as an experience of non-participation that acts as an incentive for the pursuit of deeper involvement while marginality is defined as a negative experience of non-participation that results in a sense of exclusion. Relations of inequality and hierarchy are therefore primarily established within CoP through the granting or the denial of status to would-be members.

A closer look at F/OS communities, however, shows that they are more complicated. Within the F/OS context inequality continues to be an issue long after member status is granted. The fluid boundaries of the community combined with the open character of contributions may allow for a broadened view of membership, but participation is carefully structured through the formal and informal regulation of different degrees of access in projects and resources, which result in a renewed sense of hierarchy. Although obtaining a CVS account[14] is considered by many to be the de facto sign of membership there are other levels of access, such as becoming a member of the project's administrative body. In addition, in some projects, for example KDE, having an email account with the project's domain name is a more distinguishable sign

[14] A CVS (Concurrent Versions System) is an open source version control software tool that allows developers to keep track of the changes in the code and to coordinate their work.

of membership than being allowed to commit to the CVS tree, especially to the outside world. Controlling access to an email account means control over who is seen as member of the project and who effectively has the right to represent the community. Availability of the source code does not necessarily therefore mean access to more important levels of information or decision-making. In addition, the involvement of paid developers poses significant issues with regard to the role of employment as an additional factor for structuring participation and raises the question of whether employed hackers constitute a distinctive community of practice.

F/OS projects seem then to be underpinned by practices and processes that establish a more complex and gradated idea of membership than the one implied in Wenger's and Lave model. At the same time, the reality of participation in mature F/OS communities challenges us and forces us to rethink the idea of peripheral participation and the established notion that the projects constitute a homogenous community of practice, that of programming.

The CoP approach assumes that all peripheral members are newcomers who are eager to make their way to the next level of initiation. This may not always be the case in mature open source projects. In both the GNOME and the KDE projects, the two case studies that form the core of the empirical study in this thesis, the primary communities of programmers are from time to time complemented by groups of individuals who work on non-programming tasks, such as translation, documentation, creating artwork, community public relations. Some of these individuals may be peripheral participants in Wenger and Lave's sense: programmers who make the decision to contribute to a less demanding part of the project as part of a learning strategy that will allow them to move on to programming. However, many of these peripheral or supporting tasks may be carried out by people who do not wish to move on to the centre of the primary developers' community. Indeed, many of these tasks can be seen as themselves constituting a CoP.

In order to examine these issues, this study adopts an idea that is part of the original CoP framework, but which has not been utilized in the wider literature, the notion of a *constellation of practices* (Wenger, 1998). Instead of conceptualizing F/OS communities as being founded solely on the practice of programming, in this study it is suggested that F/OS projects can be regarded as constellation of practices, a configuration of diverse but interconnected CoP. This view adopts, in part, Wenger's (1998) idea of organizations which he perceives as consisting of different CoP that give life to the institutional design. This idea is adapted to the context of F/OS, in which the institutional elements are created in response to growth and the boundaries between the different practices are not fully defined.

Lastly, the CoP perspective also underlies the view of technology adopted in this research, in terms of the way software tools and techniques are used in the context of development. In particular, the tools, processes and techniques, such as

CVS, mailing lists, IRC[15] channels, bugs databases that support F/OS developments are considered within the context of the research as the tools of trade of participants, which support engagement and communication, allow them to carry out their shared enterprise and act as repositories of knowledge and memory, allowing the information needed to be retrieved. This view of instrumental technology, of technology as artefact and process, is additional to other interpretations of technology that are explored in section 2.3.2 which outlines the Foucauldian programme for studying power.

2.2.3 CoP revisited: addressing some underdeveloped theoretical aspects

The above section has provided an overview of the CoP approach and highlighted the specific elements of the theory that are the initial building blocks for the conceptual framework in this study.[16] More specifically, this section draws attention to the way that the theory has been appropriated both by organization studies and within the F/OS literature, where the emphasis is mostly on the communal/consensual and ideological aspects of CoPs. More importantly, it points to some of the areas of the original CoP approach that have remained under-developed, namely the way that power operates within and across different CoP and the way that centrality and peripherality are experienced and constituted.

The issue of power is relevant both for relations within a community of practice that needs to regulate movement from the centre to the periphery, and for the relations across different CoP. In the first instance this means that this research examines learning not as a process of socialization of new developers into the values and practices of the hacker culture, but as an inherent element in the regulation of control and access. In the second instance it suggests that research should examine the regulation of relations across different CoP operating within a project as a necessary condition for the division of labour. The practice of power is a necessary condition for coordination and is achieved through the regulation and stratification of access and decision making and the regulation of inter-community cooperation, all of which can be expected to be established through a variety of informal and formal strategies. In some cases there may be specific policies that regulate how translators cooperate with coders. In others it might be mostly up to a specific, established contributor to act as the broker between the various communities.

The shift in perspective for considering teams that contribute in ways other than coding as CoP in their own right, and not just as the periphery of the community of coders, does not mean that the difference between centre and periphery is obliterated. Rather it might be transposed to another level, that of the relation

[15] Internet Relay Chat (IRC) is a form of real-time Internet chat and synchronous conferencing.

[16] The adoption of the CoP approach means that within the context of the research the concept of community is instrumentally defined, that is, on the basis of shared enterprise. This is not to gloss over longstanding debates on the idea of community, which remains a contested notion (Calhoun, 1980) and its transformation within the context of recent developments connected with ICTs (Rheingold, 1993) and globalization (Appadurai, 1990) but is meant to provide a firm basis for investigating the research questions.

between the various CoP. The primacy of the developers' community is often reflected in contributors' perceptions of the value of technical skills within projects. Although the contribution of non-coders is increasingly appreciated, the primacy of the technical community is constantly reaffirmed. [17] Furthermore it is often acknowledged that only a handful of peripheral contributors will ever reach developer status in the social organization of a project (Villa, 2003), (Roger, 22/08/04, Lawrence, 5/08/04)[18].The value that is attached to the work of each community of practice within a project appears to be an appropriate measure of the power that seems to be inherent in practice. There is no reason to believe that this measure is fixed. It might be reasonable to hypothesize that the work of peripheral communities becomes more important and the boundaries between communities shift as projects grow and achieve a larger user base.

2.3 Foucault and the study of power in Free/Open Source

The previous section discussed how the CoP perspective offers a suitable primary framework for studying how F/OS communities are organized. This section delves more deeply into questions of control and explains how Michel Foucault's ideas are helpful in addressing some of the deficiencies of the CoP perspective and in providing the tools for understanding many of the challenges that F/OS communities face as they grow and solidify their links with the commercial world. To achieve this several issues need to be addressed.

Section 2.3.1 provides an overview of the way Foucault's ideas have been appropriated in the context of information systems and organizational theory. This helps contextualize the discussion in section 2.3.2, which outlines his ideas on the nature and workings of power adopted within the framework of this research. Section 2.3.3 highlights Foucault's conceptualization of power relations by comparing it with the models of power that underpin Giddens' structuration theory and Bourdieu's work on the habitus and culture. Although this examination is far from exhaustive, it is intended to clarify the implications of some of the theoretical and methodological choices made in this research. Section 2.3.4 provides a synthesis of the arguments and recapitulates Foucault's contribution to the research.

2.3.1 Foucault in context: studying an emergent form of organizing

Foucault's views on knowledge and power form part of the ongoing discussions regarding the paradigms of power associated with the rise of the information society (Munro, 2000; Poster, 1990, 1984). This section highlights some aspects of this debate and draws attention to the way Foucault's ideas have been mobilized within the

[17] This view is supported by 11 interviewees.

[18] These are interviews conducted in the context of this study.

context of the study of organizations and ICTs in order to specify the point of departure for this study. The majority of studies employing Foucauldian ideas in the organizational studies literature challenge optimistic accounts regarding the empowering potential of ICTs for employees and their capacity for change. Methodologically they tend to focus on discursively analyzing managerial techniques and technologies designed to increase corporate control and expand the panoptic gaze of senior management. In contrast to these approaches, this study examines an emergent form of organizing whose dynamics of cooperation appear to differ significantly from that of established institutions. Furthermore, as indicated in Chapter 3, we also examine the structural elements of participation by looking at F/OS contributors' employment relations.

Discussions about power and control in the workplace form part of an ongoing debate about how and whether ICTs and the rise of the information society are implicated in restructuring power relations, leading to a radical shift in the way power operates in modern societies. Manuel Castells (2000a), for example, claims that the new regime of power will largely depend upon the control of flows of information in virtual space. Ian Munro (2000) adopts Castells' argument and suggests that Foucault's idea of the disciplinary society needs to be rethought in order to account for the emergence of new forms of power that are mainly associated with the control of flows of information to which the individual has access.

Within the framework of mainstream organizational and information systems many studies argue that the restructuring of power relations associated with information based technologies and the flexibility associated with post-Fordist models of organizing allow for a greater degree of autonomy and agency on the part of employees (Vallas, 1999).

Most studies that are informed by a Foucauldian perspective, however, contest this view and point to the complicity of technological systems and strategies that aim to regulate behaviour in amplifying managerial control. Wilson's (1995) study shows that many of the flexible forms of production and organization are dependent on highly centralized systems of control and disciplinary mechanisms, which, in fact, depend on more than the panoptical use of ICTs. An equally important aspect of amplifying control lies in the development of a discourse that seeks to regulate and normalize behaviour. The significance, for example, that many studies on virtual organizations place on trust, self-management and teamwork (Handy, 1995; Jarvernpaa, et al., 1998; Kasper-Fuehrera, 2001) suggests an increasing support for the internalization of control, which reminds us that the Panopticon (Foucault, 1979)[19] is

[19] The Panopticon, which Foucault perceives as a sign and metaphor for the disciplinary society was an architectural innovation designed by Jeremy Bentham at the end of the 18th century. The Panopticon was meant to introduce and establish hierarchical observation and provide the means for the integration of utility, production and control. Adopted not only in prisons, but also in factories, schools and hospitals, the model was, he argued, integrated into the architecture and geometry of these institutions the distinctive arrangements of observation and close surveillance.(Foucault, 1979)

meant to function properly, even without an observing presence. The Panopticon, which represents the essence of the disciplinary society, signifies the various control techniques, such as surveillance, detection, recording, that were developed to monitor behaviour in prisons, factories and schools and which are brought to a new level of pervasiveness and efficiency by ICTs.

Wilson's views are shared by many researchers. For example, Zuboff (1988), in her well-known book *In the Age of the Smart Machine,* shows how the introduction of information technologies in two very different organizational settings improved the transparency and therefore the regulation of workers' behaviour. Sewell and Simpson (Sewell, 1998; 1999) agree that the monitoring of computerized workflows has rendered control even more perfect and invisible. It seems that whatever strategies of resistance employees devise they are always operating in an already inscribed power/knowledge regime and cannot escape the panoptic gaze.

At the same time there are studies that adopt a Foucauldian approach, but do not reach the same pessimistic and negative conclusions. Hahnele Huthala's (2004) study entitled 'The Emancipated Worker? A Foucauldian Study of Power, Subjectivity and Organizing in the Information Age' concludes that modern corporations are characterized by a general lack of a central organizational authority with which to comply. She suggests that contemporary control operates directly, through subjectivity, rather than indirectly through a hierarchical system, and that workers are largely self-managed, no longer objects, but subjects, and the underlying principle of organizing is by negotiation of tasks, roles and processes.

Alan McKinlay (2002: 86) also demonstrated that novel organizational techniques such as 'Knowledge Management' have yet to demonstrate that they can become sufficiently durable to "accumulate the knowledge that is necessary to refine structures of power and adjust their control processes".[20] The appropriation of Foucault's ideas in the social study of ICTs and in management studies has been criticized by those who argue that his ideas have not been adequately utilized in examining crucial issues related to power and the growing use of ICTs.

Foucault's view of power suggests a more complex view of control and agency than those afforded by most applications of his ideas in the study of ICTs and in the organizational studies literature. Foucault considered the wielders and subjects of power to be equally implicated in dominant knowledge/power regimes. Wardens in prisons, for example, need to conform to the prisoners' rigorous regime and managers can be subjected to the same processes of surveillance and meticulous evaluation as their subordinates. His relational idea of power, developed in section 2.3.2, is founded on the principle that power is not exercised directly on a person, but is an action directed upon other actions that creates complex interdependencies and structures the fields of possibilities with ambiguous effects.

[20] For a comprehensive overview of Foucault's appropriation in the social study of ICTs and in management studies see (Willcocks, 2006).

However, Foucault's programme of research was not directed by a pessimistic or optimistic view of power or by a vision favouring more continuity over change. Arguably, his research agenda was focused on the blindspots of power relations across different knowledge domains and institutions. This has significant implications for the examination of power in F/OS, a domain that is dominated, as indicated in Chapter 1, by bottom-up accounts of power, evolving around the idea of practice as culture, learning as socialization, and of hierarchy as social organization. More specifically, this research examines not only discursive aspects of collaboration, but also structural aspects of participation by investigating the role of paid developers. The aspects of Foucault's work adopted in this study are elaborated next.

2.3.2 *The Foucauldian programme and power: glimpses into an alternative strategy for the study of power*

This section focuses on the theoretical and methodological elements of Foucault's work adopted within the context of this research. The focus in this study on commercialization and peripheral participation is guided by Foucault's strategy of studying the blind spots of power in institutions, which involves challenging established modes of rationalization and differentiation, and highlighting tactics for controlling access, establishing order, mobilizing participation and commitment. The research will examine discursive and non-discursive elements of cooperation and participation by: a) tracing the interconnections of bodies of knowledge that give meaning to the experience of participants; and b) mapping the material strategies, practices and connections that structure access and participation. The model of power that underlies this study's conceptualization of power in F/OS is Foucault's idea of relational power.

The first difficulty that one confronts when considering Foucault's notion of power is that he did not express his theoretical views in a systematic way that one usually identifies with a coherent body of theory. His preoccupation with power relations is a theme that spans his work, but which is not necessarily always in the foreground. In fact, as Poster (1984) points out, a common criticism is that he uses the term power in a very vague and unlocalized way, which, in addition, almost precludes possibilities of resistance. Foucault gave many explanations of his work at different stages in his life. One of his most elucidating accounts of the issues that guided his research were expressed in an interview that he gave two years before he died:

> What I have studied are the three traditional problems: (1) What are the relations we have to truth through scientific knowledge, to those "truth games" which are so important in civilization and which are both subject and object? (2) What are the relations that we have to others through those strange strategies and power relations? And (3) what are the relationships between truth, power and the self?
>
> (Foucault and Hutton, 1988:15).

Despite the vagueness in his work of the notion of power, Foucault was very explicit in suggesting concrete methodologies for examining power relations. For example, in his essay 'The Subject and Power' (1982c), Foucault indicated that:

> the analysis of power relations demands that a certain number of points be established:
>
> 1. *The system of differentiations that permits one to act upon the actions of others*; juridicial and traditional differences of status or privilege; economic differences in the appropriation of wealth and goods, differing positions within the processes of production, linguistic or cultural differences, differences in know-how and competence, and so forth[...]
>
> 2.*The types of objectives* pursued by those who act upon the actions of others: maintenance of priviledges, accumulation of profits[....]
>
> 3. *Instrumental modes*: whether power is exercised by the threat of arms, by the effects of speech, through economic disparities [...]
>
> 4. *Forms of institutionalization:* these may mix traditional conditions, legal structures, matters of habit or fashion (such as the one sees in the institution of the family[...]
>
> 5. *The degree of rationalization:* the bringing into play of power relations as action in a field of possibilities may be more or less elaborate in terms of the effectiveness of its instruments and the certainty of its results (greater or lesser technological refinements employed in the exercise of power) or, again, in proportion to the possible cost (economic cost of the means used, or the cost in terms of the resistance encountered). , The exercise of power is not a given fact, an institutional given, nor is it a structure that holds out or is smashed: it is something that is elaborated, transformed, organized; it endows itself with processes that are more or less adjusted to the situation.
>
> (Foucault, 1982c: 344-345).

The analysis of the *system of differentiations* combined with the examination of F/OS communities as constellations of practice give rise to some interesting questions in relation to the different types of knowledge and expertise recognized in the context of development and the material factors that structure participation. Analysis of the *types of objectives* guides an investigation that goes beyond the study of individual motivations to recognize the systemic influences that are negotiated within the context of projects. The examination of *instrumental modes* gives rise to the more fundamental sources of influence, the relational dynamics and restrictions that need to be taken into account within the context of development. At the same time, the examination of *forms of institutionalization* draws attention to the interdependencies forming between gift and exchange economies and to how their needs are negotiated within projects. Lastly, the examination of *degrees of rationalization* highlights the[21] tools and the techniques that support the organization and system of differentiation in F/OS projects, such as their social and meritocratic basis.

Foucault's influence on the theoretical and methodological framework for this research is not restricted to the specification of the constitutive aspects of power relations. It also affects the view of how these different elements interconnect, giving rise to concrete programmes of action that support specific rationalities, what Foucault calls 'technologies of power'. This is related to Foucault's view of relational power and his conception of how these technologies evolve and form discrete

[21] A more detailed operationalization of the research questions is provided in the chapter 3 which outlines the study's methodological framework.

'economies of power', dominant paradigms of the way power operates throughout history (Foucault, 1979).

Power, according to the relational view, is neither a zero-sum game in which different actors compete for resources, nor something that is given or exchanged, but rather something that is exercised; a force that creates complex dependencies and invites a diversity of initiations and reactions on the part of the people involved. Foucault associated this function of power with the newer forms of power introduced through disciplinary power, the dominant mode of power in capitalist society (Foucault, 1979). He contrasted his approach to the view of power inscribed in political economy, which, in his opinion, is mistakenly, preoccupied with problems of sovereignty and perceives power solely as domination. He aimed to highlight a functioning of power that was at once relational and non-subjective. Although he acknowledged that groups and individuals make conscious and intentional choices that render power relations intelligible, he denied the possibility that the broader consequences of these actions could be coordinated. The techniques and technologies of power, such as the spatial and temporal ordering of individuals, that he considered to be so important in the ordering of relations and the constitution of the self, are not the outcome of a straightforward calculation, but the generalization of, initially localized needs and demands (Foucault, 1979).

Foucault developed this notion of technology in two essays: in 'The Political Technology of Individuals' (Foucault, 1982ba) and 'The Technologies of the Self' (Foucault, 1982ab). It is worth noting that Foucault does not override the established meaning of technology as an artefact or technical process, but expands it to include the techniques that: a) "determine the conduct of individuals and submit them to certain ends or domination" that he calls '*technologies of power*' and b) those that he terms 'technologies of the self':

> which permit individuals to affect by their own means, or with the help of others, a certain number of operations on their own bodies and souls, thoughts and conduct and way of being, so as to transform themselves in order to attain a certain state of happiness, purity, wisdom, perfection or immortality
>
> (Foucault, 1982b: 225)

In drawing attention to the relational character of power, Foucault frequently used a vocabulary comprised of points, capillaries and net-like organization. In his essay (Foucault, 2002b) on 'Space, Knowledge and Power' he talks about the political significance of architecture and "conjures up an image of space as a map of relations between forces" (Allen, 2003: 70). As John Allen in his 2003 book *Lost Geographies of Power* aptly remarks, this spatially oriented view of power is not identical to the notion of power as a flow or medium frequently to be found in the work of theorists such as Castells (2003). Rather in Foucault's work power is only intelligible through its relational effects and more specifically it is "this interplay of the forces that makes it possible to extrapolate diagrams of power relations inscribed within particular institutional spaces" (Foucault, 198b: 73). This spatial view of power is consistent with methods adopted in Economic Sociology, which examine dependencies, material and cooperative ties between different types of actors, allows for a development of a topology of power in F/OS projects (see Chapter 7).

This tradition, and more specifically the body of work associated with the American Sociologist Mark Granovetter (1985), is adopted in examining the structural aspects of access and contribution, that is, the material practices that organize and order collaboration and the way structural aspects of participation, such as employment relations, and the gift and exchange economies interconnect. Granovetter examined how economic action is embedded in existing social networks and how the latter influence, hinder or promote economic cooperation. In questioning the purely social basis of participation in F/OS the present research investigates not the embeddedness of the social in the economic, but rather that of the economic in the social.

In investigating whether F/OS communities are indicative of a new mode of power supported by ICTs, this study adopts Foucault's ideas on the evolution of power relations throughout history. Like the economy of production, whose history is the basis of Marxist theory, power according to Foucault has its own economy and its own history. In reconstructing the history of the economy of power, Foucault distinguishes between three different configurations of power: a. sovereign power, in which power emanates from a single god-given source, the regent; b. disciplinary power, which is studied within the context of the formation of institutions such as prison or hospitals, where the goal of power is to produce docile bodies; and c. bio-power, the more recent expression of power, a function of government whose objects are populations and their welfare.

Foucault grounded his analysis of the shift in the predominance of different economies of power in different periods of changes in the dominant historical rationality, in the way that people thought and acted. His account of the emergence of disciplinary power is not a neat account of the gradual substitution of sovereign power by the new paradigm, but a complex and rich story of the way that different localized requirements helped develop a set of techniques that migrated from one institution to another and imbued existing practices and norms with new meaning. Each new economy of power does not simply erase the power configurations that are implicit in the previous paradigm; they transform its discourse, assimilate many of its procedures and practices and sometimes coexist with it. For example, in ***Discipline and Punish*** (1979) Foucault describes how the knowledge of organizing and distributing bodies in space and time necessitated by the adoption of the rifle and the subsequent changes in military tactics, informed industrial organization on the factory floor. This meticulous control of presence and activity, which he calls cellular power, was not, however, entirely new; it was formulated and practised for hundreds of years in monastic communities. It took off, he argues, as the dominant way of organizing as a result of the congruence of a series of complex events, some of which were linked to the rise of the bourgeoisie.

The relational, material and historical view of power supported by Foucault invites us to examine F/OS projects not as closed systems that lack historical and material connections with the offline world, but as potentially new forms of organization informed by existing practices and discourses that circulate across different domains and institutions.

This type of historical perspective and its underlying methodology, which Foucault calls 'genealogy'[22], a process that includes the analysis of an event according to the multiple processes that constitute it, is of primary importance for the examination of new forms of organization at both a discursive and a non-discursive level. For example, in Chapter 7 the importance of community discourse in F/OS projects is linked to the increasing significance of the idea of community in politics and organizations and is highlighted as an integral aspect of strategies for mobilizing participation and commitment.

In this context the notion of community is associated with a more 'natural' way of ordering relations, especially when it is founded on the, generally, undisputed principles of meritocracy, which provides the principle form of rationalization of power differentials in F/OS. This view is supported by the view of learning as an enculturation process. The predominance of social organization implicit in the idea of community, in turn, is often associated with ideas of emergence and self-organization both in practitioners' and academics' accounts, as indicated by the popularity of the cathedral/bazaar metaphor. This present study indicates how this degree of rationality and the associated strategies and practices of F/OS are combined to form three *'technologies of communities'* that are indicative of a redefinition of the relationship between sociality and economic production promoted by ICTs.

Section 2.3.3 elaborates Foucault's ideas and methodological strategies by situating them within a discussion of power and structure informed by the work of Pierre Bourdieu and Anthony Giddens.

2.3.3 The stories not told: Foucault between Giddens and Bourdieu

The transition from initial interest to theories, and then to a set of more concrete research questions, is not a linear process. Rather than providing a mere tool, the theoretical agenda also shapes the main issues for research, leading to different lines of enquiry. This becomes more evident when the alternative directions that might have been pursued are considered. In order to better situate the discussion about power in F/OS projects, therefore, Foucault's ideas on power here are compared with Bourdieu's notion of the habitus and symbolic power and Giddens' structuration theory. The aim is not to provide an exhaustive account of these comprehensive theories, but to hint at their different agendas in order to highlight the theoretical and methodological choices in this study.

[22] The method was developed by Foucault in an interview entitled 'Questions of Method' and combines genealogical and archaeological modes of inquiry (see (Foucault, 2000a).

David Swartz (1997: 286) points out that: "Like Foucault, Bourdieu searches for deep structures of cultural and social life that are linked to power". In trying to relate individual action and social structure, Bourdieu formulated the notion of habitus, a concept that signifies the individual's tacit knowledge about the social world, which allows him or her to interact in it. The habitus functions somewhat similarly to discourse. It sets structural limits to action and generates perceptions and practices that correspond to the process of socialization through which it is instilled in individuals. Bourdieu conceived the habitus: "both a system of schemes of production of practices and a system of perception and appreciation of practices" (Bourdieu, 1989: 19). In attempting to explain the persistence of stratified social systems of hierarchy and domination Bourdieu used the notion of symbolic power, which he defined as the power of "world making" (Bourdieu, 1989: 22) . Symbolic power is a key element in political struggles since the relations that are established at this level reflect and reproduce objective relations of power.

What kind of agenda would Bourdieu's theoretical formulations set for the examination of power relations in the context of this research? One could begin to understand the relations between the different groups on the basis of symbolic power and start to think about differences in their social standing within the project in terms of cultural and symbolic capital, the different kinds of resources that the participants draw from. On this basis one could ask why certain forms of capital, that is, forms of knowledge and types of contributions are more recognized than others, or translated, in Bourdieu's terms, as symbolic capital. At the same time, the tensions and challenges that arise through the involvement of companies, could perhaps be conceptualized as part of a process of 'field formation', the gradual institutionalization of F/OS within a broader arena of opposing forces. There are of course other possibilities and this agenda is only indicative of some of them. In the F/OS literature, for example, Bourdieu has been mobilized to explain the organization of relationships within the development community as one that is based on relationships of kinship (Zeitlyn, 2003).

So why is this line of enquiry not pursued here? First, because Bourdieu's conception of power, which he considered predominantly as a medium through which social groups mobilize their collectivity, is considered somewhat limited. In conceptualizing relationships between the different groups of participants on the basis of relations of domination, one loses the scope for analyzing the different kinds of interdependencies that develop between them. Secondly, Bourdieu emphasized patterns of continuity more than change. Foucault's framework, on the other hand, provides the possibility for highlighting both the continuity in change and the change in continuity.

Giddens' structuration theory and his notion of distantiated power also represent a view of power that is distinctive from that elaborated by Foucault. Structuration is defined as a social process that involves the recursive interaction of individuals and institutions. The theory of structuration acknowledges that human actions are enabled and constrained by structures, rules and resources, that allow the reproduction of social systems across time, but at the same time recognizes that these structures are a result of previous actions. Structuration theory is primarily a theory of agency and one of its most important propositions is the duality of structure, which implies "that the rules and resources drawn upon in the production and reproduction of social action are at the same time the means of system reproduction" (Giddens, 1984:19). Giddens suggests that structural properties express forms of domination and power and that institutions effectively represent those properties with the greatest time expansion. Therefore, the ability to structure and control relationships across time and space is one of the most significant properties and expressions of power. In Giddens' theory this is achieved through the mobilization of allocative and authoritative resources that are stored in certain locales. Allocative resources are usually identified with material resources, such as land, goods and technology. Authoritative resources refer to authority or control over the ways according to which social life is organized.

A research agenda informed by these ideas might investigate how coordination is achieved by examining F/OS projects as virtual organizations. Following Giddens' idea of distanciated power this approach would pinpoint a project's power locales and turn attention to the mechanisms and channels that disseminate the decisions made to the wider community. At the same time, structuration theory might provide an appropriate framework for studying mechanisms of reification within CoP that make up the projects, on the one hand, and the way that informal processes and tactics become formalized and adopted as official community procedures and policies on the other.

According to John Allen (2003: 47), Giddens' view of power implies that: "power generated in one part of a distanciated network is transmitted intact across it. As an understanding of power, therefore, it is possible to read such distanciated networks merely as conduits for the transmission of all kinds of organizational and institutional ability". This is typical of the 'networked' conception of power, which although it provides us with a rich vocabulary of reach, scope and action 'at a distance', "the flow of power through the networks does none the less appear remarkably routine" (Allen, 2003: 60).

Giddens has also been criticized for formulating a view of power in which the agency perspective is dominant (Clegg, 1989). This is reflected by his somewhat under-developed conception of structure. John Thompson (1989) specifically draws attention to the theoretical inadequacy of defining structure as consisting of rules and resources. Although Giddens seems to provide a suitable framework for the study of power in F/OS, his conceptualization of structure and the way that power is disseminated seems to provide a less nuanced account of power than the one that can be woven by adopting a Foucauldian approach.

Bourdieu, Foucault and Giddens developed frameworks for understanding one of the most central issues of social science, the relation between agency and structure.

Their work provides a rich tapestry of ideas and concepts that can be used in order to study the way that structures and hierarchies emerge and the scope for individual and collective intervention. Each theory, however, implies a distinctive theoretical and research agenda and thus frames and limits the research questions that can be addressed. Although the possibility that Bourdieu's or Giddens' ideas could be relevant in understanding the findings of this research is not rejected, it is argued that Foucault's understanding of power and power relations steers the original interest of the research into shaping more interesting research questions and provides a more compelling theoretical and methodological framework for examining them.

2.3.4 Foucault revisited: towards a new economy of power?

In organizational studies Foucault's work has been mobilized in order to examine the use of ICTs and the development of managerial techniques designed to increase corporate control and expand the panoptic gaze. These studies tend to focus on the abilities of new technologies and techniques to reify existing, disciplinary and calculative structures and often challenge their proclaimed abilities to induce positive change in the life of employees.

This study departs from this line of research in that it uses Foucault's idea for studying an emergent form of organization, F/OS projects, which appear to defy both the way that traditional organizations operate and the dominant economic rationality. In the same way that existing Foucauldian studies question managerial discourses regarding the empowering potential of ICTs, the strategy here aims to highlight the blind spots in F/OS projects regarding the dynamics of participation and collaboration. This involves examining discursive as well as structural aspects of participation and collaboration in F/OS projects with the aim of critically examining prevailing explanations regarding how they are organized.

An important argument made in the following section and one that arguably has not been adequately addressed by existing studies is that F/OS development does not happen in a void. The way that people perceive and rationalize their participation in F/OS work may be connected with their experiences and perceptions of work in more traditional environments. Dominant views of how F/OS is organized and why it is successful might also be connected to discursive strategies and practices adopted by or developed in other institutions. More importantly, as indicated in the relevant literature F/OS communities have important and multifaceted connections with the exchange economy. Combined with the idea of F/OS projects as constellations of practices, Foucault's idea of relational power is mobilized as the means of tracing specific programmes of action and of mapping the dependencies that develop between different groups of community members.

2.4 *Enacting structures: Goffman, face-work and the negotiation of image in Free/Open source*

As indicated in section 2.2, the CoP perspective sets the general framework for this investigation of the research questions and Foucault (section 2.3) provides the guiding principles and notions according to which power within in F/OS is studied. Goffman's ideas concerning the order of social interaction complement these approaches by drawing attention to the communicative aspects of F/OS. More specifically, this research employs his notion of face-work to study the collective dimensions of identity and representation. Although Foucault clearly indicates that power relations cannot be reduced to or be seen to emerge from relationships of communication,[23] within the context of this study, processes of face-work are considered as one of the arenas in which power relations are played out. Face-work may be an important aspect not only of advocacy and the mobilization of participation within and across different groups of contributors, but also of the management of tensions related to growth and commercialization.

Section 2.4.1 provides a brief overview of the way Goffman has been appropriated within the context of organizational and CMC (Computer Mediated Communication) studies. Section 2.4.2 focuses on the way that his ideas are mobilized within the context of this study. Section 2.4.3 deliberates on the complications of combining Foucault's and Goffman's ideas. Section 2.4.4 summarizes the main arguments presented in this section.

2.4.1 *Goffman in context*

The points of contact and departure with studies that have mobilized Erving Goffman's work in a context comparable to that of the present research are considered in this section. Within Internet studies[24], Goffman's theories have been used mainly in the study of web pages. Some work in organizational studies has adopted his ideas of face-work and impression management as tools for the examination of issues such as conflict and corporate image and this applies also to the field of CMC.

Within the context of Internet Studies, Goffman's work seems to have been exclusively mobilized for the study of personal homepages. (Miller, 1995; Miller and Arnold, 2001; Walker, 2000). Researchers have adopted the notion of self-presentation to examine how individuals negotiate and validate their identities through the construction of personal web pages. Hugh Miller (1995) constructs a crude categorization of personal web pages and attempts to highlight the significance and the evolution of their expressive means and resources. In a more recent study,

[23] See (Foucault, 1982c) pp.337.

[24] The term Internet studies refers to the interdisciplinary field concerned with the investigation of the socio-cultural and economic implications of the Internet. One of the most established researchers associations of the field is AOIR (Association of Internet Researchers). For an account of the evolution of the field see (Wellman, 2004).

conducted in cooperation with Arnold (2001), he examined representations of women academics on the web. This time the focus was on the challenges that women academics face in presenting an individual identity within an institutional web page framework. Katherine Walker's (2000) survey of personal web pages indicated that there are two main types of personal web pages: those that individuals construct to maintain relationships formed apart from the Internet and those that people create to contact other denizens of cyberspace. She argues that the motives for creating web pages are related to differences in overall Internet use and in authors' conceptualizations of the Internet.

All three studies support the view that, in addition to supporting the construction and negotiation of identities, personal Internet home pages constitute a means of communication. For example, Hugh Miller (1995) maintains that in mounting a homepage individuals open themselves to the possibility of interaction. However, even if we accept the validity of his argument, the fact remains that web pages are a rather static means of communication. Goffman has developed an array of concepts and ideas suitable for exploring the more interactive and dynamic aspects of communication.

Two such ideas, which are also adopted within the context of this research, are the notions of strategic interaction and impression management. Within the wider field of organizational studies these ideas are mainly used to explore the strategies and tactics of corporate and institutional politics. Robert White and Dallas Hanson (2001) exploit Goffman's concepts in order to examine how a multinational forestry and manufacturing company used its annual reports to control the construction of its reputation. Soren Christensen and Ann Westenholtz (1999) conceive the board of directors of a Danish company as a theatre ensemble staging a performance for the stakeholders and the wider environment.

Internet studies that adopt Goffman's ideas turn attention to the mediating framework of new technologies and the possibilities that it offers for the presentation of the self. As such they form part of the branch of CMC studies that examines the relationship between offline and online social life and the frames that shape and regulate it (Slater, 2002; Turkle, 1996). Organizational studies, on the other hand, that adopt Goffman's ideas focus on the strategic, more explicitly political, aspect of communication (Christensen and Westenholtz, 1999; White and Hanson, 2001). Although the question of face-work and impression management is inextricably linked with identity, this research focuses mostly on the dynamic and strategic aspect of communication.

In examining these issues this research takes into account the framing effects of different communication media. The notion of frame, which could be considered as part of the function of mediation, refers to the rules and conventions that Goffman perceives as part of the organization of experience which helps to define a situation (Goffman, 1974). F/OS contributors use a wide range of different communication tools and frameworks: they meet face to face in Linux expos, project conferences at hacker workshops, and PGP[25] key signing parties. They use real-time IRCs, exchange emails or post them on public mailing lists. Although this research does not examine in detail the media ecology that supports F/OS development, it does draw attention to the way contributors adjust their tactics according to the way they perceive their audiences, the various contexts of interaction and the dynamics of the situation.

2.4.2 *Goffman and F/OS: negotiating image and sustaining participation*

Goffman's theory of social interaction, the way that individuals interact within a given social structure, was influenced by the Chicago School's symbolic interactionism which attributes great importance to the interaction between symbolic verbal and non verbal resources in understanding social life (Smith, 1999). Although Goffman acknowledged the usefulness of such a framework, he considered it to be too abstract and elaborated a theory that took into account both the determinative role of frames, occasions and associated semiotic codes and the capabilities of individuals to improvise creatively within these structures.

Among Goffman's favoured metaphors for social life were the game, the ritual and the drama. In the context of each metaphor he drew attention to different, often complementary and overlapping, aspects of communication and interaction. The notion of the ritual for instance, was more frequently associated in his work with sustaining the moral order, whereas the notion of the game emphasized the strategic, calculative role of interaction. The idea of drama often combined the two since, as Goffman indicated, a successful performance that is being staged as part of a strategic move usually involves acknowledgement and acceptance of the social order (Goffman, 1969b).

The notion of face-work describes a special kind of performance. Goffman defined face: "as the positive social value a person effectively claims for himself by the line others assume he has taken during a particular contact. Face is an image of self delineated in terms of approved social values" (Goffman, 1969a:5). A line is "a pattern of verbal and non verbal acts which he expresses his view of the situation and through this his evaluation of the participants, especially himself" (Goffman, 1969a:5). A person 'maintains face' when he or she succeeds in conveying the kind of information that is consistent with his or her face. In contrast, a person 'loses face' when he or she is either out of face, when he/she fails, that is, to present a line that is expected of him or her in a particular social encounter, or when he or she is in the 'wrong face', "when information is brought forth in some way about his social worth which cannot be integrated, even with effort, into the line that is being sustained by him or her" (Goffman, 1969a:5).

Face-work then is a term that designates the actions that a person takes in order to 'save face'. The fact that a person usually develops an attachment to his or her

[25] PGP (Pretty Good Privacy) is an encryption method that ensures privacy and the secure identification of the sender.

chosen face, a perceived positive image combined with an inability to control every flow of information that can be conveyed, make, according to Goffman, every social encounter a commitment; a commitment for a person to conduct him or herself in such a way as to maintain both his/her face and the face of the other participants. However, not every encounter is based on the principle of mutual considerateness. In addition to defensive face-saving practices, such as the tactic of keeping away from sensitive, potentially embarrassing topics, Goffman also points to the aggressive use of face-work, where the encounter is constructed as a game in which each person tries not to lose face, while at the same time, tries to score points against his/her opponents, making them lose face.

Goffman extended the concept of face-work to team performances. The term refers to any set of individuals that co-operate in staging a single routine. Team performances are characterized by reciprocal dependence, which links individuals together. Goffman, however, did not consider these two types of performances to be independent of one another because the effectiveness of individual performances usually depends on the projected agreements of others.

Team performances within F/OS appear to take on many different dimensions, but this study focuses mostly communication across different groups of contributors; and handling relations between companies and the community. The notion of team face-work is employed to explore tensions related to the increasing growth and commercialization of the F/OS projects that appear to test the limits of the commonly held view of F/OS development. The first dynamic is associated with increasing participation, the second with the involvement of companies and the use of the project's applications in production environments. As communities grow and strengthen the links with the commercial world a number of issues arise. Stability may become as important as innovation and there may be more pressure to involve people with different skill sets in order to streamline processes and raise the profile of the project.[26]

The notion of face-work is a useful analytical tool because it allows us to take into account both ritualistic and strategic, game-like aspects of interaction. The ritualistic aspect may be related to how the order is sustained through the acknowledgement and the reification of community values, such as the need for inter-project cooperation, the appreciation of user feedback, the acknowledgement of the significance of coders and the projection of a professional image to the outside world. The strategic aspect may be connected to the need to introduce new processes or improve existing ones by encouraging the cross-project cooperation of groups belonging to different CoPs and by enlisting the help of coders. It is interesting to note that Goffman did not seem to think of morality and manipulation as two opposing concepts. As Ann Branaman (1997) points out, this is especially clear in the

[26] See Freeman's distinction between radical and incremental innovation (Freeman and Soete, 1987)

case of face-work, where strategic considerations need to take into account and demonstrate that they comply with the designated moral order.

Although this study focuses mostly on the collective, team dimensions of face-work, it also takes into account some of its individual expressions. At an individual level face-work is likely to be mainly associated with how individuals stage their different roles. This is especially interesting for the groups of participants that have vested interests in the development of the project, such as employed contributors. It may be that sponsored programmers do not experience their roles of community member and employee as incompatible. This may depend on how convinced they are of the image they project, or the performance that is front stage. Goffman can sometimes be read and applied cynically in this regard.

However, there are less cynical interpretations that are not inconsistent with Goffman's framework. These take into account the efforts of employed community members to handle or preempt tensions by negotiating special work agreements that endow them with a considerable degree of independence. Although Goffman emphasizes that people take on different roles in different settings, he also acknowledges the continuity of a person's biography: "Each artifact and person involved in a framed activity has a continuing biography, that is a traceable life (or the remains of one) before and after the event and, and each biography ensures a continuity of absolute distinguishables, that is selfsameness" (Goffman, 1974:87).

People tend to experience their different roles as part of their continuing biographies. Longstanding community members may have woven their experiences of participation into their identities and the positive social values associated with them. One hypothesis worthy of exploration is that receiving pay for participation is not experienced by the participant or the community as a contradiction, especially if the former has had long involvement in the project. This does not mean that there are no tensions arising from the need to accommodate community and commercial interests. The relationship between commercial actors and the community, which is often considered beneficial for all parties, may lead to conflicts of interest at both an institutional and a personal level.

The concept of face-work contributes considerably to the study of power in F/OS projects in two ways. First, by revealing how different actors try to accommodate the tensions arising through the growth and the gradual institutionalization of F/OS projects at both a personal and a collective level. Second, by contributing to the mapping of the different configurations and dynamics of power that are present in F/OS communities, by paying attention to the kinds of considerations being taken account of by certain teams and certain types of actors as they try to achieve their goals and manage their roles. The concept has its limits with regard to the examination of power relations, however. Some of these limits have methodological implications and do not enable us to address the issue of access. However, we need to demonstrate how Foucault's and Goffman's ideas, and more specifically, the notion of face-work and relational power, can be meaningfully combined.

2.4.3 *Foucault and Goffman: between face-work and relational power*

This section traces some of the parallels in Michel Foucault's and Erving Goffman's works. The comparison of their respective theoretical and methodological frameworks does not extend to every aspect of their work, but revolves around the central concepts that are adopted within the context of this research, mainly Foucault's ideas about relational power and the conceptual models and material practices that constitute it, and Goffman's concept of face-work. The discussion is facilitated by reference to Ian Hacking's (2004) ideas concerning the compatibility of Foucault's and Goffman's theoretical and methodological agendas. Although the study draws from different aspects of their work than those that Hacking incorporates in his work, Hacking's contribution is significant because it demonstrates that these two theorists can be meaningful combined.

Hacking (2004) sees the theories developed by the two scholars as complementary, especially in the context of his work, which focuses on the interactions between classifications of people and the people classified. More specifically, Hacking sees what he calls Foucault's 'top-down' archaeological approach, which is directed at entire systems of thought, as complementing Goffman's 'bottom-up' approach which focuses more on everyday interactions and on how, through them, the forms of discourse revealed by Foucault become parts of the lives of ordinary people.

However, Hacking points out that he is more interested in completing Foucault with Goffman than in helping to deal with the limitations of Goffman's blind spot by using Foucault's ideas. Goffman's blind spot lies in a lack of understanding of "how the institutions he described came into being, what their formative structures are" (Hacking, 2004:278). In studying how power relations are constituted and maintained within F/OS communities this research combines other aspects of Foucault's and Goffman's ideas. However, like Hacking, this study does not mobilize their respective theories symmetrically and, similarly, this research is more influenced by Foucault than Goffman.

In contrast to Hacking's work, this research draws more from the Foucauldian notion of 'genealogy' than that of 'archaeology'. Archaeology was a method developed by Foucault in order to help us explore the emergence of discourses, of bodies of knowledge, that define the parameters of what can be said and what can be seen as a set of social arrangements. One of the basic premises of Foucauldian thought is that people do not think or act according only to some universal rules of logic, but also according to specific historical rationalities (Foucault and Hutton, 1988).

While Foucault employed archaeology as a way to explore these types of historical rationality mainly at the level of discourses, he formulated the concept and method of genealogy as a specially designed strategy for the study of power. Although genealogy shares some similarities with archaeology, since it also takes into account the significance of discourse, it pays equal attention to the materiality of power as expressed through specific actions and strategies. Foucault (1980: 85) specifies the difference and the relation between the two methodsin the following terms: "If we were to characterize it in two terms, then 'archaeology' would be the appropriate methodology of this analysis of local discursivities, and 'genealogy' would be the tactics, whereby on the basis of the descriptions of these local discursivities, the subjected knowledges which were thus released would be brought into play".

The idea and method of genealogy are important because they are closely connected with the idea of relational power. By problematizing the present and the 'natural' function of institutions, Foucault seems to be highlighting the subtle interdependencies between existing and emergent discourses and strategies and creating an alternative account of the way power operates. Although he principally drew attention to the totalizing aspects of power, Foucault firmly argued that the programmes of totalitarian institutions could never be fully realized. This is because of all the different, and often competing strategies that are being deployed within their context. Foucault described what is happening in prisons as a 'witches brew' compared to the beautiful Benthamite machine. Relational power does not only signify mutually reinforcing forces. Within the context of this research, relational power also helps to highlight the variety of, often competing, discourses and strategies that may be deployed within each context of action. Such a framework leaves room for a theorist such as Goffman to provide a context for understanding how people may take advantage of this situation in order to promote their interests and the values to which they subscribe.

This may be especially true of F/OS communities that have not assumed a stable institutional form. This makes the mapping of the different discursive and non-discursive elements that underlie power relations in F/OS more challenging. Face-work has a double function here. On the one hand, it reveals how people try to negotiate and position themselves within discourses and practices that make-up the space of projects. One the other hand, it allows for an indirect mapping by highlighting the kind of concerns that shape people's performances. Within the context of this research face-work is considered as both as an expression of and evidence of relational power.

2.4.4 *Goffman revisited: face-work and the study of power*

This section so far has focused on the communicative aspects of F/OS and discussed how Goffman's notion of individual and team face-work can capture some dimensions of relational power. Face-work is an especially helpful idea since it is an expression of both the strategic and ritual aspects of interaction. The positive values associated with F/OS set up the context of interaction and are both the limits to and the object of strategic manipulation.

The mobilization of face-work for the study of power relations is not unproblematic. The multi-layered structure of F/OS development and organization and the regulation of access create a series of challenges for the researcher who is often restricted to an examination of 'on stage' performances. It is highly unlikely that unrestricted access to all the levels of the community, including what goes on in the boards of their administrative organizations or those of the companies that participate in the projects, would mean that the researcher would be in the position to observe individuals having all their cards on the table. Indeed one question that springs to mind when reading Goffman is whether there is a space or a time when people stop performing. However, it would undoubtedly be useful to compare performances conducted at different project levels, both back stage and front stage, and more or less mediated. This was possible on only a few occasions in the conduct of this research.

In addition, Foucault argues that communication (if it produces any image at all) can often present us with a distorted image of power relations. Face-work, especially in its most mediated form, can help us to trace some of the threads in power relations, but the concept cannot enable us to trace all of them. It can only add significantly to our understanding of power relations in F/OS projects if a certain degree of knowledge about the actors, their strategies and the configuration of power exists. As it has been indicated this weakness is exemplified by the limitations of studies employing social network analysis (see Chapter 1, section 1.3.2).

Moreover, the emphasis given within the context of this research to the strategic aspects of communication can be misleading if its significance is overestimated. At the end of his book on strategic interaction Goffman (1970:135) acknowledges that communication does not always work as a game of tic-tac toe:

> Typically the offended party is neither compelled fully by honor nor governed fully by cool strategic design. He does not use his turn to make a move; he gets by with half actions. Instead of commitments and enforcements he provides assurances and resentments. Instead of moves mere expressions. To translate this gestural realm into entirely into strategical equivalents is to violate its regrettable nature; we end up by making sustainable imputations of complex play to persons who aren't quite players and aren't quite playing.

This is confirmed by F/OS project mailing list archives which reveal a multitude of single message threads and unfinished discussions.

The next section introduces the notion of 'meshwork' which builds upon Foucault's idea of relational power to conceptualize the network of practices and discourses that underlie F/OS development.

2.5 *Meshworks, Cathedrals and Bazaars*

This section refers back to the concepts of the bazaar and the cathedral to indicate the historically contingent character of their appropriation in the context of F/OS studies and introduces the idea of relational meshwork, which in this study is used as an analytical tool to disentangle the complex web of discourses and practices that underpin F/OS development.

In many ways the social sciences signify the quest for the appropriate metaphor, the search for the form that will succeed in translating the messiness of the world into a somewhat familiar and finite structure. As previously indicated two of the most pervasive metaphors in the case of F/OS, the 'cathedral' and the 'bazaar'. As shown by the persistence of the emergent character of F/OS social organization and the dichotomy between top-down and bottom-up, the expressions of power that these two metaphors represent have pervaded the thinking of practitioners and academics alike. However, history and social science indicate that these two organizational forms are underlined by much more complicated processes than Brooks (1995) and Raymond (2001a) envisaged. So why have the notions of the cathedral and the bazaar acquired their specific meaning within the context of software development and what does that imply? More importantly, however, is there a way of examining power relations in F/OS while taking into account both the discourses that inform the reception and understanding of this phenomenon and the structural factors that influence how these discourses are actually realized?

Perhaps the dichotomy between the perception of software development as the building of a cathedral and its understanding as the outcome of an organic process is connected with the emergence of software engineering as a technoscience (Quintas, 1993). Within this context Frederick Brooks' seminal work *The Mythical Man Month* (1975) can be seen as reflecting what had been perceived as the need for the systematization of the processes that underlie systems development. Brooks' conviction of the importance of top-down design and the instrumental role of a master programmer in overseeing its implementation was based on the idea of the significance of form, the primacy of the whole over the parts. This is essentially how Brooks understood the building of cathedrals, as the careful adherence to a master plan which kept at bay the forces of the entropy of communication that led to the fall of the Tower of Babel.

In his book *Masons, Tricksters and Chartographers* David Turnbull (2000) challenges Brooks' conceptualization by providing a very different account of the way that Chartres cathedral was built. Turnbull questions the idea that a building as complex as Chartres could not have been built without a detailed plan and an architect by suggesting that the construction site functioned as a large laboratory where talk, tradition and templates provided the necessary basis of coordination. More specifically:

> This [the building of Chartres] was possible in the absence of fully articulated structural theory, specified design or plans, or even common measure, because the builders developed ways in which their local and tacit knowledge and their disparate practices could be combined and transmitted to other sites in the form of skills, geometric methods and templates. This constituted a tradition of common solutions and skills in which theory and practice were integrated and no strong distinctions were made between science and technology.(Turnbull, 2000:81).

If the building of cathedral, according to Turnbull, is a relatively unstructured practice, the bazaar seems to imply a more structured process than Raymond envisaged. In his (1979) seminal study 'Suq: the bazaar economy in Sefrou' Clifford Geertz provides a unified account of the way a Moroccan bazaar is organized both as a cultural and an economic entity. Geertz suggests that the bazaar has a distinctive social form that regulates practical relationships between buyers and sellers, professionals and laymen, apprentices and masters. The bazaar of Sefrou is first and foremost: "an enormously complicated, poorly articulated, and extremely noisy communication network" (1977:125). According to Geertz the bazaar is able to function without the presence of a conscious or explicit organizing influence because replication and specialization act as successful substitutes for scale. A key aspect of its structure is what Geertz calls a "delicacy of discrimination", a combination of a complex division of labour and man-in-his provenance discrimination, which, moreover, depends on a system that favours partnership over employer-employee relationships.

These examples are meant to indicate that the relation of top down and bottom up expressions of organization that became idealized in the metaphors of the cathedral and the bazaar are more complex than either Raymond or Brooks envisaged. Rather than representing a mere dichotomy, the two forces stand in a dynamic relationship to each other. More importantly, however, as these two studies indicate, the reception of the bazaar and the cathedral on Brooks and Raymond's terms is historically contingent.

The way that F/OS has been constructed as an experience is also not neutral in value. The prevalence of bazaar over other metaphors applied to F/OS, such as Alan Cox's more modest perception of F/OS development as a town-hall meeting or Raymond's initial intention of modelling the open source process over the classical ideal of the agora is, if not accidental, not without implications. Moreover, the idea of the bazaar appears to mobilize older discourses that have been formalized in traditional management contexts and echoes discussions about emergence, self-organization and complexity, which are frequently underlined by a distinctive neo-liberal agenda (Best and Kellner, 2000). In other words, the world in a sense was ready for F/OS and the bazaar and F/OS as the bazaar. At the same time, however, the F/OS model has an impact on real people and real organizations. Companies that are involved in F/OS development frequently employ volunteer developers and make adjustments to their processes. These might be marginal, but are incorporated nevertheless. Whether they will ever amount to significant changes is a question that probably only time will answer.

In examining the network of discourses and practices that underlie F/OS projects the idea of a meshwork is mobilized. The notion builds upon the idea of meshwork developed by Manuel de Landa in his (2000) book *1,000 years of non-linear history*. De Landa uses the notion of a meshwork to talk about the bottom-up function of markets and contrasts it with the idea of top-down control that is inherent in the function of bureaucracies. This idea is not introduced to establish another metaphor which reproduces the dichotomy of the cathedral and the bazaar, but is employed in order formulate a guiding concept that enables a conceptualization of the organizing dynamics of development and allows the dynamic interdependence of discursive and structural dimensions of power and control to be captured. F/OS is subsequently conceived as a meshwork of interconnected discourses, material and idealized practices which weave together the foundation for understanding and the context of action that connects people and institutions through power relations that wear, reinforce and add threads to the tapestry.

2.6 *Conceptual Framework*

This research aims to examine the F/OS model of development by highlighting the dynamics and power relations that underlie the organization of F/OS projects. To achieve this, a conceptual framework is developed that examines F/OS projects not as closed systems that lack historical and material connections with the offline world, but as emerging types of organization connected with existing practices and discourses.

This research adopts the CoP perspective such that F/OS project communities are conceptualized as consisting of different constellations of practices where learning serves as a factor controlling access and participation. This opens up the possibility of examining the experiences of participation and collaboration between different groups of contributors, mainly, volunteer and employed programmers, senior and new developers, and non-programmers. An examination of the role of paid coders provides insights into how the gift economy is embedded in the exchange economy, and an examination of the participation of new programmers and non-programmers enables an elaboration of the notion of peripherality. The selection of these groups was motivated by Foucault's strategy of challenging the prevalent rationality of how relations of power are experienced, justified and organized (see Chapter 3). In the case of F/OS this implies a view of F/OS projects as homogenous, volunteer communities built around the practice of programming.

In conceptualizing power relations between and across the different CoP that participate in a given project and the commercial actors with stakes in the development, the conceptual framework mobilizes Foucault's ideas on the historical and relational character of power and adopts his methodology on how power relations are examined. This includes the adoption of the strategy of genealogy and the examination of the structural elements of participation. The investigation of the structural aspects of participation also benefits from research in Economic Sociology that focuses on the idea of embeddedness[27] (Uzzi, 2001, 1996). The adoption of the idea of a relational model of power does not imply that the relations to be examined are symmetrical, but is meant to acknowledge and map the multiple interdependencies and structures that are being developed within the context of projects that shape a complex topology of participation. This topology is expressed by the idea of a meshwork.

Goffman's idea of face-work is considered as a means of exploring and extending the assumptions about relational power. Its property of combining both ritual and game like aspects of interaction make it an appropriate tool for examining how people try to negotiate and position themselves within a meshwork of power. One the other hand, the concept of face-work is expected to contribute to an indirect mapping of the threads of power, by highlighting the values and concerns that people take into account in individual and team performances.

In short, the conceptual framework builds the following elements:

- An approach to the study of F/OS projects as constellation of practices;
- A conceptual framework for studying power derived from Foucault's ideas on how power operates that problematizes the novelty and self-sufficiency of the bazaar model of development by drawing attention to its blind spots, which are specifically associated with the dynamics of commercialization and peripherality.
- A framework for studying the way structures and dynamics are enacted and negotiated provided by Goffman and the notion of face-work.

This conceptual framework guides the investigation of the central research questions that were introduced in Chapter 1 and set out at the beginning of this chapter:

- What are the dynamics that underlie F/OS software projects?
- How are power relations between various actors constituted and maintained at different levels of interaction?

These questions will be examined by pursuing the following lines of investigation, mainly:

- How are relations and practices between different groups of contributors organized and maintained?
- How are the ideas of peripherality and centrality experienced and constructed?
- How does learning structure access and participation?

[27] For a detailed discussion of the idea of embeddedness, its connection with other sociological and economic lines of research, its appropriation and critiques see 'Editor Notes on Further Reading Mark Granovetter, 'Economic Action and Social Structure" in (Granovetter and Swedberg, 2001), pp.74-76).

- How do the gift and exchange economies intertwine and in particular: What impact does commercialization have on development and community dynamics and how are the resulting tensions accommodated within a community's structure and processes?
- How is the idea of community perceived, contested and confirmed among the various groups of contributors and the developments associated with its evolution?

The overarching theoretical question the research examines is:

- Is the F/OS model of development indicative of a new form of power relations supported by ICTs?

These questions and the methodology for their study are elaborated in the next chapter.

3 Methodology

3.1 Chapter overview

This chapter outlines the methodology for this research. The methodology relies on a three-step research design that combines quantitative and qualitative methods, which examine discursive and structural aspects of participation and cooperation in F/OS. Section 3.2 presents the main elements of the methodology and the way the principal research questions are operationalized. Section 3.3 presents the rationale for the selection of the primary case studies - the KDE and the GNOME projects.

Sections 3.4, 3.5 and 3.7 outline in more detail the components of the research design: the interviews, the nested case studies and the quantitative analysis of the GNOME Foundation and KDE e.V. members, and of the two projects' maintainer networks. In each case attention is drawn to the rationale, the implementation and weaknesses of the specific design and its connection with the conceptual and operational framework and the other empirical components of the research. Section 3.6 outlines the framework for analyzing the qualitative findings and discusses the type of discourse analysis employed. The framework for analyzing the quantitative findings is described in section 3.8. Section 3.9 summarizes the main elements of the research design.

3.2 Overview of the research design

This section introduces the basic elements of the research design which aim to disentangle the power relations that underlie the dynamics of participation and collaboration in F/OS projects. This involves examining the discursive and structural dimensions of participation and collaboration in the GNOME and KDE projects. This section also indicates how the research questions are operationalized.

The collection and analysis of the findings was organized in three phases. Phase 1 consisted of 23 interviews conducted with the sample, which comprised four groups of participants: novice developers, volunteer and paid developers, peripheral contributors (mainly translators and documenters). Six of the interviewees were involved at the higher administrative and technical levels of the project (board members, members of the release team, translator and documenter coordinators). The interviews examined how different types of contributors reflected on a similar set of issues, and elicited information regarding different aspects of development and cooperation.

Phase 2 consisted of three nested case studies, the GNOME Bounty Hunt, the case of the Gstreamer/Fluendo and the KDE Quality teams and examined aspects of commercialization and peripheral participation that had emerged from the analysis of interviews conducted in Phase 1, in more detail.

Phase 3 consisted of a quantitative analysis of the patterns of contribution of GNOME Foundation and KDE e.V. members, and of the two projects' maintainer networks. This element of the research aimed to test specific hypotheses regarding the contribution patterns of volunteers and paid developers and of peripheral contributors and programmers. The instrument used for data collection on the GNOME Foundation and KDE e.V. was a questionnaire that was distributed by email to members. The maintainers of the various project modules were identified through an online search.

During these three phases the following F/OS community events were attended:

- GUADEC (GNOME Users and Developers Conference) 2003 Dublin, Ireland (16-18 June).
- GUADEC 2004, Kristiansand, Norway (28-30 June).
- GUADEC 2005, Stuttgart, Germany (29-31 May).
- FOSDEM (Free and Open Source Developers European Meeting) 2004, Brussels, Belguim (21-22 February).
- KDE Akademy 2004, Ludwisburg, Germany (21-24/08).
- KDE Akademy 2005, Malaga, Spain, (22/08-4/09).
- Gstreamer/Fluendo summit 2005, Barcelona, Spain (17-21/2).

The significance of these events for the research was twofold. First, they often led to the identification of and introduction to interviewees. Secondly, observing how F/OS contributors interacted provided significant insights into the various aspects of community life and development.

Table 3-1 indicates how the research questions are addressed. The principal questions (first level operationalization) presented in the Chapter 1 were further operationalized at the end of Chapter 2. This chapter breaks these questions into sub-questions that correspond to the second level of operationalization. Table 3-1 indicates in which phases of the research the questions were examined. Justification for the research questions is further discussed in the presentation of the interview thematic guides, the survey questions and the protocol for the online identification of the project maintainers. First, justification is provided for the selection of the two primary case studies: GNOME and KDE.

RESEARCH QUESTIONS				PHASES		
Principal Research Question s		First level operationalization	Second Level Operationalization	1	2	3
Is the F/OS model of development indicative of a new form of power relations supported by ICTs? What are the dynamics that underlie F/OS software projects? How are power relations between various actors constituted and maintained at different levels of interaction?	Peripherality	How are the ideas of the peripherality and centrality experienced, and constructed?	What are the frameworks of meaning and experience associated with participation in each sphere, how are different skill sets and contributions evaluated between the examined groups?	☑	☑	
			Is the idea of peripherality negatively constructed?	☑	☑	
			Are peripherality and centrality connected with employment status?			☑
		How does learning structure access and participation?	What are the barriers to access for new contributors?	☑	☑	
			How is learning connected with other fundamental project dynamics (i.e. meritocracy, volunteer basis of participation, gift culture)?	☑	☑	
		How are relations and practices between different groups of contributors organized and maintained?	How is cooperation between coders and non-coders structured?	☑	☑	
			How are the different group agendas, including those of paid developers prioritized?	☑	☑	
		How is the idea of community contested, confirmed and perceived among the various groups of contributors?		☑	☑	
	Commercialization	How do the gift and exchange economies intertwine?	How are the boundaries between companies and the communities managed and negotiated at the meso and micro level?	☑	☑	
		What impact does commercialization have on development and community dynamics? How are the resulting tensions accommodated within projects?	What is the role of paid developers in development and community life?	☑	☑	
			How do new types of interfaces set between the community and companies, such as bounty hunt contests influence project dynamics?	☑	☑	
			What areas do paid developers contribute to and maintain?			☑

Table 3-1:Operationalization of research questions and phases

3.3 Selection of primary case studies: the GNOME and KDE projects

The diversity of F/OS projects, the variety of their governance structures, the ideological differences within the movement and the presence of commercial sponsors, significantly complicated the project selection process. The F/OS movement can be seen as constituted of dozens of jigsaw pieces, each deserving of its own analysis. Like fractal patterns, however, many of issues regarding power and organization are also reproduced within the F/OS projects themselves. The projects finally selected for the research were KDE and GNOME, which are mature projects in both a technical and a communal sense and have developed significant ties with the commercial world.

There were two reasons for deciding to study mature projects. The first was because of the need to achieve a depth of analysis that would have been impossible if only immature projects were included. Most importantly, however, as became apparent in the early stages of the research, was the different quality of the insights from communities that have been involved for longer in commercialization and the issues surrounding it. The experiences and discourses embedded in the institutional forms as well as in the informal processes and practices are characterized in this case by a reflexive quality that is often lacking in immature projects. As indicated in section 3.5.1.2, the case of Gstreamer/Fluendo serves largely as a control study, since this project is just beginning to form its ties with the commercial world.

Both KDE and GNOME aim to provide a complete desktop environment for free operating systems, such as Linux and Unix. Both projects also have administrative organizations whose primary mission is to represent the community to the commercial world. In GNOME this is the GNOME Foundation and in KDE it is KDE e.V. Many consider KDE to be a more popular application to work on, and the project is said to have a relatively lax policy concerning new members. Studying projects with close relationships to each other benefited the research in many ways. For example, it challenged the view widely held among the developers that the requirements of the code dictate the organization of the project. Moreover, pondering on the differences between the two projects provided a very useful way of eliciting insights about how a community is perceived, and distinguished from others.

For projects with such a long and, at times, tumultuous history[28] mutual comparisons were unavoidable. KDE's dependence on the Qt, a toolkit that was produced by a Norwegian based software company called Trolltech and which was not

[28] GNOME was launched in 1997 an KDE in 1996.

originally available under a free software licence, provoked many questions in the early stages of the development of the project. Many thought that this choice endangered the future of the community and undermined the independence of the project. Trolltech might at any time refuse to release newer versions of Qt to the KDE developers. The problem was resolved with the founding of the KDE Qt Foundation. Although Qt currently includes a closed source development licence option for commercial development purposes, Trolltech has signed a contract with the administrative and legal body that represents the community, KDE e.V., which commits them to releasing all future versions of Qt as Free Software.

GNOME was developed partly in response to the need for a desktop that had stronger connections to the free software world than KDE. In contrast to the majority of early KDE developers, many of whom, according to a KDE interviewee (Roger 24/08/04), had developed on a Windows platform, most of the founding GNOME hackers were involved in Linux kernel development. The two communities now have many links with the commercial world. Their products are included in most commercial Linux distributions and they are bundled with the software of large commercial players, who also sponsor several community developers. Some of the most important companies involved in GNOME include Novell Inc., which a few years ago bought Ximian, a company that was created by the GNOME founders, Miguel de Icaza and Nat Friedman, and Sun Microsystems Inc. and RedHat Inc. KDE is the default desktop of Novell Inc./SUSE and Mandrake, two important Linux distributions.

KDE and GNOME are not among the most technical projects, at least on terms of all aspects of participation. The fact that they are addressed to the average user means their participation is open to people with relatively few, and ideally no technical skills. This is not the case for projects such as Apache Server, whose users are primarily system administrators. It could be said that GNOME and KDE are somewhere in the middle between projects requiring a high degree of technical knowledge, and/or a high degree of domain knowledge, that is, knowledge related to the specific area of the projects' application (for example, scientific applications). This characteristic may influence the dynamics between the centre and the periphery of these projects, since it defines the areas of expertise and is a factor that needs to be taken into account in generalizing the study's findings.

The choice of translators and, to a lesser extent, documenters and usability experts, as the periphery groups to be studied was based on a similar rationale. They were chosen over developers working on accessibility, and over artists who create graphics for the various applications, because they represent two different areas of expertise and levels of technical skill sets (higher in the case of documenters and lower in the case of the translators) and interact more closely with coders than other peripheral actors, such as artists.

Although these two factors, the technical profile of the projects and the characteristics of the chosen peripheral groups, limit the generalizability of the findings at the level of specific interactions, they do not prevent important insights from being drawn about the kinds of dynamics that might underlie such relations, for example, the way that expertise and contribution are valued and defined in a constellation of practices.

3.4 *Phase 1: Interviews*

This section introduces the first element of the research design, which consisted of 23 semi-structured individual interviews with a range of contributors from KDE and GNOME. The interviews examined how contributors occupying different positions in the community reflect on a similar set of issues and helped to build an understanding about various social and technical aspects of the development process. Section 3.4.1 draws attention to sampling issues and the themes covered by the interview guides. It allows us to see how the questions elaborated in section 3.2 were applied in the interviews. Section 3.4.2 highlights the links with the conceptual framework and reflects on the strengths and weaknesses of the interview approach in the context of the study.

3.4.1 Sampling and thematic guides

The interview respondents were sampled on the basis of their involvement in different communities of practice. More specifically, the selection process aimed to construct a balanced sample consisting of volunteer and paid hackers, peripheral and central contributors, new and experienced developers. Table 3-2: presents the split between the various categories. A detailed breakdown of the sample, which includes a profile for each interviewee, is included in the Appendix (pp. 235-240).

Paid developers	6
Volunteer coders (experienced)	5
Peripheral contributors (non-coders)	7
Novice developers (newbies)	5

Table 3-2:Interviewee sample

The research process revealed that there is a significant overlap between these categories and also highlighted the need for their further elaboration. For example, all the peripheral contributors were volunteers. As the quantitative analysis in Chapter 6 indicates, this is by no coincidence.

The interview guides[29] followed Foucault's suggestions for investigating power relations (see Chapter 2, section 2.3.2). Three of Foucault's analytical categories: *system of differentiations, types of objectives, degrees of rationalization* were directly addressed by the interview guides. The instrumental modes and forms of institutionalization were underlying analytical themes that related to many different questions. Specifically, the themes examined in the interviews were:

a) The **system of differentiations and types of objectives** adopted by different groups of contributors in relation to each group. This category of questions investigated *how new and experienced developers, peripheral contributors and paid developers think and talk about each other and their aims.* Each group was asked to assess the contributions and roles of the others and to explain some common differentiations made within the context of the projects (for example, what do the terms core developer and core contributor signify?). An attempt was made to circumvent the issue of respondents simply engaging in face-work by asking them to reflect on some widely accepted views. For example, within open source it is often believed that paid contributors are paid to do only the boring 'stuff' that the volunteers would never be concerned with. In the case of peripheral contributors it is often suggested that relationships have greatly improved and that earlier in the life of the projects they were under-appreciated (Why was that, what has changed? Are there still problems?). This aspect of the research provided valuable insights into the underlying rationality of the system of differentiation interviewees established between the different types of contributors.

b) *The learning processes and the barriers to learning* that underlie the process of becoming a full member of the community and of moving from the periphery to the centre. Because of the importance attributed to learning and the efforts that are being made to better organize the learning process, learning was considered **as part of the degrees of rationalization** in the way power operates. Interviewees were asked about the difficulties they had when they first joined the project, and the difficulties that they still face. They were prompted to reflect on the issue by questions regarding the importance of having senior developers acting as mentors within the community and about whether projects could better organize the learning process and what that would involve. As is indicated in Chapter 5, which focuses on peripheral participation, the challenges

[29] For the interview guides see the Appendix (pp. 228-230).

involved in the learning process are closely associated with justification of the meritocratic basis of participation and often express one of the ways power operates (**instrumental mode).**

c) **Degrees of rationalization** (other than learning). Examination of this aspect of development *investigated various formalized and semi-formalized control and structuring processes*, including, the organization of releases, access to the CVS system, etc., and the way that they reflect various agendas and project dynamics. For example, interviewees were asked for their views on the role of the administrative bodies and the release teams and about whether F/OS projects could benefit from more or less centralized management and/or from more formalized processes.

d) *Experiences and expectations associated with participation.* This group of questions prompted contributors to reflect upon the **systems of differentiation** underlying F/OS development from another perspective. This time contributors were asked to reflect on they way that they perceived their participation in F/OS. Interviewees were asked how the experience of cooperation in the project compared with working in more traditional work environments and were asked to discuss the differences and the similarities between the two contexts of work and to describe how their initial expectations compared to the reality of participation. The comparison between the different contexts of work provided valuable insights into the instrumental mode underlying F/OS model of development.

e) *Evolution and professionalization of the community.* This group of questions examined how the involvement of companies and certain aspects of the professionalization of the community are perceived and accounted for. This theme aimed to explore how the influence of the exchange economy is rationalized, whether its aims and strategies were perceived as conflicting with those of the gift economy **(types of objectives)**, and to examine perceptions regarding the co-option of the community by corporate actors.

In addition, the interview guides included questions designed to explore issues specific to certain types of contributors. For example, paid developers were asked to indicate how they managed to balance their potentially conflicting roles as employees and community members and peripheral developers were prompted to reflect on their relationship and cooperation with coders.

3.4.2 Connection with the conceptual and operational frameworks and links to other elements of methodological framework

This part of the research aimed to test some of the initial research hypotheses regarding issues of power, and investigate systems of differentiation, instrumental modes, forms of institutionalization, types of objectives and degrees of rationalization associated with peripherality, centrality and commercialization in F/OS development. Table 3-3 summarizes the relationships examined and connects them with the second-level operationalization presented in Table 3-1.

Phase 1						
	Interview Themes		Connection with 2nd level Operationalization (Table 2)	Underlying analytical themes		
Systems of differentiation	How do different types of contributors think and talk about each other? How do different contributors evaluate different skill sets, levels and types of expertise? Is participation in F/OS perceived as similar or radically different from other spheres of activity?	What other types of distinctions do contributors make at the level of community members? What do experienced coders expect from new members?	How are the ideas of peripherality and centrality experienced and constructed? What are the frames of meaning and experienced associated with participation in F/OS? Are they different from traditional contexts of work?	Instrumental Modes	Forms of institutionalization	How is the emerging system of differentiations supported and rationalized?
Degrees of rationalization	What are the main control and structuring processes underlying F/OS development?	How do these processes reflect the various project agendas? What difficulties do new contributors face when starting to contribute?	How are relations and practices between different groups of contributors organized and maintained? How does learning structure access and participation?			How are these processes justified? How is power exercised?
Types of objectives	How are different contributor agendas prioritized and accommodated within the context of development? How has the community evolved?	How do paid contributors manage their roles as employees and community members?	How do the gift and exchange economies intertwine? How kind of impact does commercialization have on development and community dynamics? How are the resulting tensions accommodated within projects?			How are different priorities and agendas hierachized in the context of community development?

Table 3-3 Connection of the study of power and Phase 1

Starting with a rough conceptualization of the centre and periphery dialectic, the interviews in the first phase of the research aimed also to examine the validity of the CoP perspective. The complexity of the system of differentiation that emerged through the very early stages of the research offered some major insights into the power that seems to be inherent in practice, and also problematized the way that practice that lies at the heart of the community is defined. For example, can volunteer and employed coders be considered to be members of the same community of practice simply because they contribute code?

The most significant problem that emerged through the first phase of research was the relative lack of focus and context that the interviews provided. This lack of context which is one of the more important drawbacks to this method, substantially limited the investigation of the central issues, such as the way in which participation and exclusion are experienced (Flick, 2002). The strategy of regarding large-scale mature projects as the main units of analysis, which was pursued in the case of these interviews, therefore becomes problematic when such a wide range of issues is examined. The adoption of the method of nested case studies aimed to address this weakness by providing a centre of gravity that helped consolidate the interview data by investigating in more depth some of the issues that emerged in Phase 1.

3.5 *Phase 2: Nested Case Studies*

This section presents the second element of the research design, which consists of three nested case studies that were selected within the context of the two mature projects. Each case study exemplifies an aspect of the F/OS process of development related to the main issues of the research and acts as a focus of meaning that allows the research questions to be investigated more thoroughly. Section 3.5.1 outlines the method and highlights the rationale for the selection of the three nested case studies, which is theoretically driven. These are:

a) The **GNOME Integration Bounty Hunt**: an initiative of Novell Inc. that was supported by the GNOME Foundation, which offered monetary rewards to programmers that provided fixes for certain bugs (faults in applications).

b) **Gstreamer/Fluendo**: Gstreamer is a F/OS project with close ties to GNOME based on which a company called Fluendo SL was built.

c) **The KDE Quality teams**: an initiative aiming to coordinate various peripheral activities.

The background to each case and its links with the research are also presented, and examples provided of the cases that were considered and rejected. Section 3.5.2 recapitulates the links with the conceptual framework of the research and highlights the way they complement the other elements of the research design.

3.5.1 Rationale for selection, background information & research design

The choice of the term 'nested' rather than 'embedded', which is used by Yin (2003) in his book on case study design, reveals an important difference in the way that a "case studies within case studies" research design is perceived here. Although Yin recognizes the possibility of an embedded, single case study research design, that is, a research strategy consisting of many, case studies contained within a single case study, he reserves the term "embedded" for a design that aims primarily at replication. Each case in this design is selected according to Yin (2003: 47) "so that it either (a) predicts similar results (literal replication) or (b) predicts contrasting results but for predictable reasons (theoretical replication)".

This was roughly the logic that guided the selection of KDE and GNOME, the two mature, large-scale project case studies. The sampling of the nested case studies, however, is based on a different rationale, according to which the sub-cases were selected in order to examine in greater depth certain aspects of commercialization and peripheral participation that emerged through the first phase of the research. These cases are not holistic studies however, in the sense that not every aspect is taken into account, nor does each provide answers to all the research questions. In order to illustrate more clearly the way that the sub-cases contribute to the research this section outlines their background and the way that they relate to the study of commercialization and peripherality.

3.5.1.1 The GNOME Bounty Hunt Contest

In addition to investigation of the role of paid hackers, the GNOME Bounty Hunt exemplifies tensions arising out of the coexistence of different kinds of power and strategies: those of the community, the Foundation, and the companies involved in project development. Its specificity lies in that it does not express the dynamics of commercialization either at the interpersonal level, such as the involvement of paid developers, or at a high institutional level, such as the GNOME Advisory Board. The bounty contest is representative of the new types of interfaces that are developing between the community and companies that directly affect the development process and agenda.

The GNOME desktop integration Bounty Hunt was an initiative of Novell Inc. and more specifically of the group of employees that previously worked for Ximian. According to the official announcement, the contest aimed to recruit new developers and to contribute to the integration of various projects, making the desktop a more coherent and complete user experience. The bounty hunt consisted of a number of small, concrete projects, each associated with a cash prize. The initiative appears to run counter to the popular beliefs regarding the efficiency of the F/OS model of development and its reliance on volunteer labour.

This nested case study is based on six semi-structured, individual interviews, three of which are with company employees involved in the contest, one with an ex-employee and GNOME board member who followed the contest closely, and two with contestants. It also utilizes related online material. The study investigated the background to the contest including its organization, and highlighted its successes and difficulties. These were associated both with the contests' organization and its acceptance by the community. As the analysis of findings in Chapter 4 indicates, this study offered significant insights into the dynamics that shape the development agenda and the way that community resources are perceived and managed.

3.5.1.2 Gstreamer and Fluendo SL

Gstreamer is a F/OS project that aims to provide a complete multimedia platform for F/OS. Compared to GNOME and KDE, Gstreamer is less technically developed and features a significantly smaller developer community. On 20 April 2004 a company called Fluendo SL was launched to provide applications and services associated with Gstreamer's multimedia platform. The company quickly recruited a number of key Gstreamer contributors.

The scope of this nested study is the broadest of the five and could be developed into a full study of an immature project, but because of the need to bound the research we do not develop it here. The project offers an opportunity for examining how commercialization gradually affects the social dynamics and development processes. Unlike the case of other companies and F/OS projects, whose relations are now more or less established and mediated through the Foundation, the case of Gstreamer and Fluendo SL provides an opportunity to observe the processes associated with these developments as they unfold.

The nested case study consisted of six interviews, three with ex-Gstreamer volunteers now Fluendo SL employees and three with Gstreamer volunteers, and observation at a volunteer-employee meeting that took place at the company's headquarters in Barcelona in February 2005. The study, the findings from which are included in Chapter 4, highlighted the challenges and difficulties associated with the transition from a purely volunteer project to one dominated by a corporate player, and examined how the boundaries between the company and the community were managed.

3.5.1.3 KDE Quality teams

The KDE Quality teams initiative aimed to centralize and formalize certain peripheral aspects of the project, such as activities related to communication, promotion and usability, and to provide an entry points for new developers by maintaining updated task lists and developing relevant documentation.

More specifically the initiative aimed to:

- Help new users and developers with programming and documentation
- Help with Bugzilla management
- Help with user interface and general testing
- Improve communication and promotion.

The KDE Team responded through a community effort, in an organized and professional manner, to issues that emerged as the project grew. Although the coding aspects of the projects have always been considered to be underlined by the highest professional standards, other aspects of the community are frequently regarded as rather amateur. Efforts to address issues such as marketing on a more professional basis have not always been successful. KDE League, an initiative of the KDE e.V., died after a series of accusations about transparency regarding the handling of funds donated by companies for public relations purposes.

This nested case study consists of four interviews with members of the initiative and highlighted the challenges inherent in coordinating peripheral aspects of participation, such as the difficulties of mobilizing programmers' involvement.

3.5.1.4 Other nested cases that were considered

During the process of selection of the three nested case studies two other processes were considered, but eventually rejected. The first was the case of an Indian software company called Wipro Technologies Ltd. Wipro Technologies Ltd was commissioned by Sun Microsystem Inc, a company with significant interests in the development of the GNOME platform, to work on certain parts of GNOME. Although Sun Microsystems Inc. has a large team of its own cooperating within the GNOME community, there were things that they simply did not have time to complete. Unlike Sun's developers, Wipro Ltd coders' performance was assessed on the basis of metrics specified in their contract. Initially it was believed that Wipro Ltd would be not be visible to the community and that contact with them would be mediated by Sun Microsystems Inc..

Despite the fact that many Wipro Ltd developers were coached by Sun employees hacking on GNOME, their presence was initially felt considerably by the community. Wipro Ltd developers would post on mailing lists and the IRC asking for clarifications and advice. The community which usually had to deal with a only few novices at a time, felt its resources stretched by the need to respond to the requests of 12 new developers. More importantly, however, Wipro's programmers had no previous experience in working on an F/OS project. After overcoming the first difficulties Wipro Ltd employees became as invisible as they were originally meant to be. It is unclear what parts of GNOME they are working on and they have not become active community members.

A case study on this group would have highlighted the difficulties of adjustment and acclimatization for a community of coders with no prior experience of F/OS and with a very different agenda from the rest of community developers. Although the case of Wipro Ltd never evolved into a stand-alone study, it emerged as a theme during the first phase of analysis. In the end, the insights afforded by the case were folded into the study of commercialization.

The second nested study would have involved the study of the Welsh translation team. It would have focused on how peripherality is experienced and on the way in which relations between the communities of coders and translators are managed. This study would have provided insights into how participation is structured and understood at the fringes of the community. As in the case of Wipro Ltd, the first phase of the study provided enough data on the translation process such that examination of these issues at the level of a specific team was considered redundant.

3.5.2 Connection with conceptual and operational frameworks and links to other elements of the methodology

In considering the place of the nested case studies in the research design it is useful to imagine them as voices in a fugue guided by two keys. The major key represents the Foucault's suggestion about the method that should guide the study of power relations: "Rather than analyzing power from the point of view of its internal rationality, it consists of analyzing power relations through the antagonism of strategies" (Foucault, 1982c: 329). The second, the minor key, represents Goffman's ideas about face-work, about the way individuals, depending on how they perceive their position within a given situation, will try to defend the positive, social values that are associated with themselves and their communities of practice (Goffman, 1969a).

The nested case studies of the GNOME Bounty Hunt contest, the Gstreamer/Fluendo case, and the KDE Quality teams act as foci of meaning that allow in-depth investigation of the issues of commercialization and peripheral participation. As the analysis of the related findings will indicate this enables a more thorough investigation of the function of face-work, that is, how actors positioned at different strategic choices defend and justify their choices and agendas. It also enables the connection between commercialization and peripherality to be traced.

The following section outlines the method for analyzing the qualitative findings.

3.6 Analysis of qualitative data

3.6.1 Thematic Analysis

The qualitative data analysis was a two-step process. First, the interview data were organized thematically. This resulted in the organization of the interview materials in distinctive conceptual unities, which enabled the identification of persistent themes and associations. These thematic units and established associations were analysed in more depth through discourse analysis. The first step produced coding frames[30] presented in Figure 3-1 and Figure 3-2 which indicate some of the relationships between codes that are later explored discursively. The software package Atlas Ti was used for coding. As the two figures indicate, the findings, following the operationalization schema, were organized on the basis of the themes of commercialization and peripheral participation. The interviews conducted within the context of the nested case studies were coded using this scheme but were also examined as self-contained units.

The coding framework followed the operationalization schema of the interview guides presented in Table 3-3, but also included codes that emerged through the analytical process. One emerging code was 'Innovation and Maintenance', which later proved significant for understanding the tensions between the gift and exchange economies. The codes that are common to both coding frames, such as 'Community and Project Evolution', provide the first clues on the relationship between commercialization and peripherality. The first level analysis of the findings is presented in Chapter 4 - analysis of commercialization - and Chapter 5, which discusses the findings related to peripheral participation.

[30] The code book is included in the Appendix (pp. 231-234).

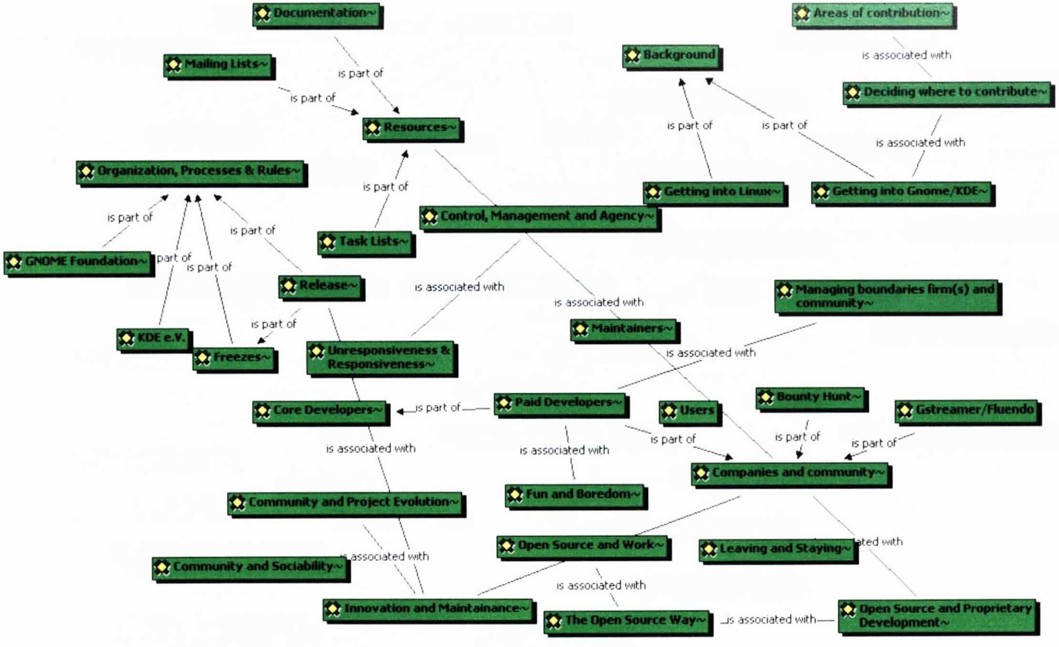


Figure 3-1 Codes on commercialization generated through Atlas Ti

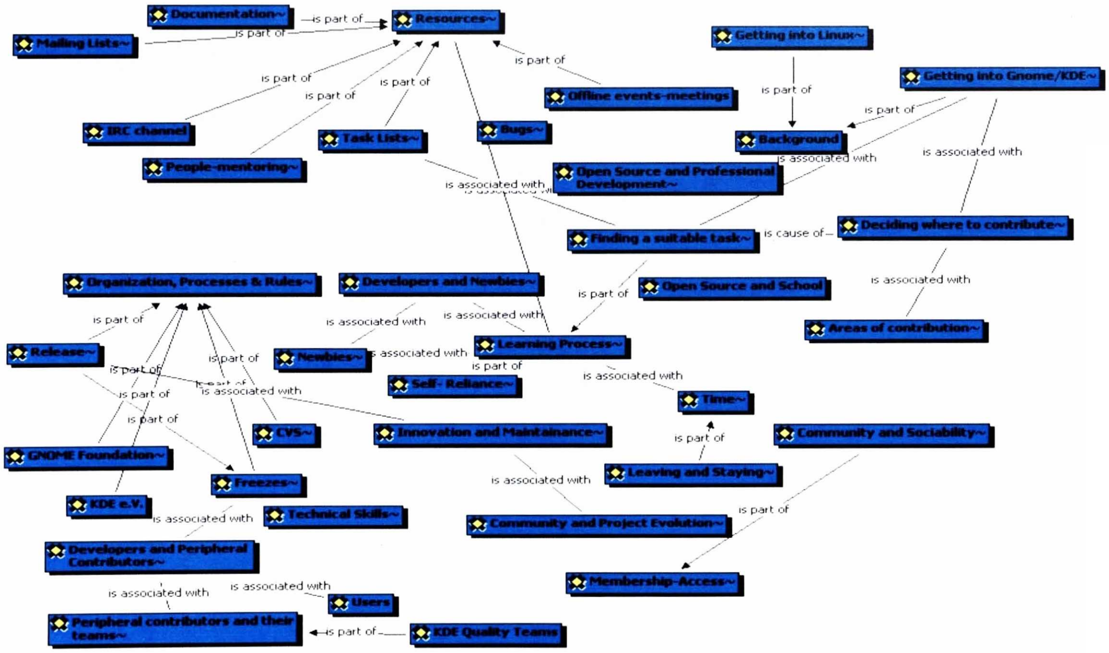


Figure 3-2 : Codes on peripheral participation generated through Atlas Ti

3.6.2 *Discourse Analysis*

Foucault conceived of discourse as the conceptual terrain in which knowledge is formed and produced (Foucault, 1981). His conception of discourse as the dominant rationality that structures what can be conceived of and expressed in any given era and within any field of activity is inextricably connected to his ideas on the way power operates. What differentiates Foucault's conception of discourse, and the corresponding mode of its analysis, from other definitions of discourse is that he defines it primarily as a *form of practice* (Hook, 2001). In contrast to modes of discourse analysis, such as those that emphasize the textual, linguistic dimension of discourse (Potter and Wetherell, 1987) and which tend to reduce it to a series of statements and narratives, Foucault invites us to investigate discourse in tandem with the material practices that support it and are produced by it. As he indicates in his essay 'Questions of Method' (2002), his aim was to examine the relationship: "between a 'code' which rules ways of ruling (how people are to be graded and examined) and a production of true discourses which serve to found, justify and provide reasons and principles for these ways of doing things" (Foucault, 2002a: 230).

This relationship between discourse and practice, which is adopted in this study, is aptly described by Julian Henriques and his co-authors:

> In practice, discourses delimit what can be said, while providing the spaces-the concepts, metaphors, models, analogies, for making new statements within any specific discourse...The analysis which we propose regards every discourse as the result of a practice of production which is at once material, discursive and complex, always inscribed in relation to other practices of production and discourse. Every discourse is part of a discursive complex; it is locked in an intricate web of practices, bearing in mind that every practice is by definition both discursive and material.

(Henriques, et al., 1984:105-106) as quoted in (Kendall and Wickham, 1998:41)

At the same time, the examination of discourse also involves its situation within a historical context. In *Discipline and Punish* (1979) Foucault demonstrated how various practices and modes of rationalization were transformed and combined to form the emerging disciplinary power. As indicated in Chapter 2, section 2.3.2 this forms part of the method of genealogy.

Based on this definition of discourse, the discursive analysis of the meshwork of power underlying F/OS development involved analysis of the way people talk and think about the constitutive elements of the F/OS model of development (such as the *system of differentiations* that supports the distinctions made between the different types of contributors) and investigation of associated practices, rules and processes that substantiate issues of access and status. For example, the priority assigned to the work of coders is associated with the seasonal character of peripheral contributors' work (see Chapter 5, section 5.4.2.). The investigation of the interconnections between different types of categorizations and their associated practices highlighted those higher level discourses and programmes of action associated with the F/OS model of development, which within the context of this study have been termed 'technologies of communities'. In line with the aims of the genealogy method this aspect of the study aimed to provide a way of peeling back the layers of meaning that render a framework of reference and meaning and its associated regimes of practices 'natural' (Foucault, 2000a).

3.6.3 Field work observations

As indicated in section 3.2, seven community events were attended during the course of the study. These events were open to the public and my identity as a researcher was made known to those who inquired about my involvement in the project because, firstly, it would have been very difficult to sustain the identity of a genuine participant, given the degree of technical knowledge required to do so, even as a user. Secondly, given the importance F/OS contributors attach to transparency, had I tried to hide it, any future revelation of my true identity might have jeopardized access to interviewees.

These field trips did not, therefore, constitute participant observation, that is, they did not provide descriptions from the perspective of an insider. Rather, they gave the opportunity to conduct many of the interviews and enabled the development of a sense of context regarding the dynamics of collaboration and the primary concerns of the GNOME, KDE and Gstreamer communities, as reflected in the issues discussed at these events. For instance, the visible presence of corporate actors during the first community conference that I attended, GUADEC 2003, at an early stage in the research, prompted reflection on the role of paid developers in the development process and on how this role is accounted for within the community. In line with the tradition of ethnography, this process generated field notes. Although the observations recorded in these notes do not amount to *thick description* (Geertz, 1979) of the examined communities, they provide a richer picture of community dynamics than could have been built by observing online interactions alone. These reports, an example of which can be found in the Appendix (pp. 241-246) framed the interview data analysis.

3.7 Phase 3: Quantitative examination of patterns of contribution and maintenance

3.7.1 Connection with conceptual and operational frameworks and links to other elements of the methodology

Phase 3 was designed to examine certain issues that emerged through the first two phases of the study. This section begins, therefore, with a discussion of the links between this phase of the research and the previous two.

The third element of the research design involved a quantitative analysis of patterns of contribution and maintenance among GNOME Foundation and KDE e.V. members and among the two projects' maintainers. More specifically, this phase of the research was designed on the basis of a preliminary analysis of the qualitative findings (Phase 1, 2) to test certain hypotheses regarding patterns of contribution and maintenance among volunteer and paid coders, programmers and non-programmers. The hypotheses are discussed in detail in chapter 6, which presents the results of this third phase:

1. Paid developers are more likely to contribute to critical parts of the code base.
2. Paid developers are more likely to maintain critical parts of the code base.
3. Volunteer contributors are more likely to participate in aspects of the project that are more geared towards the end-user.
4. Programmers and peripheral contributors are not likely to participate equally in major community events.

This section provides the rationale for studying these patterns within the context of these populations, and discusses the questionnaires that were used to gather data from GNOME Foundation and KDE e.V. members and the protocols that were developed in order to identify the maintainers.

Despite its limitations, that are discussed in Chapter 8, the study highlights important patterns of contribution that complement the qualitative aspects of the research. Unlike other large-scale quantitative studies that examine employment patterns across F/OS projects, this study provides some interesting insights into how employment can affect patterns of participation within the context of the same project. Another innovative aspect of the study concerns the advanced understanding that it involves regarding the significance of contributions. Unlike studies that examine patterns of contribution at the level of individual modules by mining CVS data, this study develops an elaborate meta-coding scheme, which allows us to differentiate between critical, important, secondary and peripheral areas of activity. Lastly, compared to most Social Network Analysis studies that infer patterns of collaboration from patterns of communication, this study provides a more direct mapping of the relations examined.

3.7.2 Studying participation and contribution among GNOME Foundation and KDE e.V. members

This section provides the rationale for studying patterns of contribution and community involvement at the level of GNOME and KDE's principal administrative bodies the GNOME Foundation and KDE e.V. and outlines the criteria for their eligibility for membership. This helps to draw out the implications of the relations in these two organizations and to frame the generalizability of the findings. Secondly, the questionnaires that were distributed to the members are discussed and details are given regarding the data gathered.

3.7.2.1 GNOME Foundation and KDE e.V.: background and membership eligibility criteria

The GNOME Foundation http://foundation.gnome.org/ was founded in 2000. It comprises the Board of Directors, the Executive Director, the Advisory Board and the GNOME Membership Committee. The Foundation coordinates GNOME releases, specifies which projects belong in GNOME and acts as the official voice of the community. At the time of the study in June 2005, it comprised 335 members. Membership, which is renewed every two years, is given to individuals who have contributed non-trivial improvements to the project. The contributions of applicants are confirmed by a long-term GNOME contributor or maintainer and the status of member is granted by the membership committee. According to an interviewee who was on the membership committee the definition of 'a non-trivial contribution' has gradually broadened. The membership guidelines that were endorsed by the Foundation in 2002 specify that:

> A non-trivial contribution is any activity which contributes to the development of the project at a level significantly above that expected of a normal user or fan of GNOME.Examples of non-trivial contributions include hacking, bugfixing, extensive testing, design, documentation, translation administration or maintenance of project-wide resources, giving GNOME talks at conferences and community coordination such as bugzilla or release management. Any activity, such as advocacy or submitting bug reports, must substantially exceed the level of contribution expected of an ordinary user or fan of the project to qualify an individual for membership in the Foundation. (http://mail.gnome.org/archives/foundation-announce/2002-October/msg00003.html, last accessed 10/2006)

A Foundation member can stand for membership on the Board of Directors, vote in Board of Director elections and issue or approve a referendum.

The K Desktop Environment e.V. (KDE e.V.) http://www.ev.kde.org was registered as an association under German law in 1997. Its mission is to represent the KDE community in financial and legal matters. The organization was initially created to represent the community in the KDE Free Qt foundation (see section 3.3). The articles of association specify that:

> An active membership will be granted if suggested by a member and supported by two other active members, if the general meeting decides to grant it or if a simple majority of the active members is obtained by means of an internet vote. The main criterion for granting membership should be the candidate's commitment over a longer period of time and the contributions he/she made in order to fulfil the Association's aims (http://ev.kde.org/corporate/statutes.php, last accessed 18/05/06),

KDE e.V. members can stand for membership of the Board of Directors and are expected to contribute towards the aims of the association which involve:

> the promotion and distribution of free desktop software in terms of free software, and the program package "K Desktop Environment (KDE)" in

> particular, to promote the free exchange of knowledge and equality of opportunity in accessing software as well as education, science and research. (http://ev.kde.org/corporate/statutes.php, last accessed 18/05/06).

The process of KDE e.V. membership is, therefore, slightly more complicated than in the case of GNOME. This is perhaps one of the reasons why, despite its longer presence KDE e.V., at the time of the research in April 2005, had 113 members while the GNOME Foundation had 335. In Chapter 6, which presents the quantitative findings, some additional factors contributing to this difference in membership numbers are discussed, including how the two organizations are perceived by their respective contributors.

An examination at the level of the two organizations means that the population for the research consisted of experienced and recognized project contributors who in addition were interested in participating in the higher institutional and administrative aspects of the project.

3.7.2.2 Questionnaires and Data gathered

The instrument us for this aspect of the study was a questionnaire that was distributed via email to all listed members of the GNOME Foundation and KDE e.V. Some additional questionnaires were distributed at the GNOME Foundation meeting in GUADEC 2006 (GNOME Users and Developers Conference) in Stuttgart, Germany. The survey yielded 199 responses from Foundation members (152 via email and 47 at the conference, which corresponds to a 59.4% response rate) and 63 from KDE e.V. members (55.7% response rate).

The questionnaire included questions relating to:

- Areas of contribution
- Affiliation
- GUADEC or KDE e.V. Akademy attendance
- City of residence
- Additional optional remarks pertaining to the respondents' participation

Question 1 asked respondents to identify:

a) Their areas/modules of contribution. For GNOME Foundation respondents the information was listed online and respondents were asked merely to confirm the list.

b) Their **main** area of contribution/module. This was defined as the area that respondents had been most active in during the six months prior to the survey.

Question 2 asked respondents to identify whether at the time of the survey they were:

a) employed by a company or an organization to hack on GNOME or KDE?

b) employed by a company or organization to work on a F/OS project other than GNOME or KDE?

Question 3 enquired whether they had participated in the last three main project conferences, GUADEC (2003, 2004 and 2005) for GNOME and KDE Akademy (2003, 2004) for KDE. Data regarding KDE e.V. members' participation in the KDE Akademy 2006 in Malaga, Spain, were provided by the conference organizers.

Question 4 asked respondents to specify where (which city) they were living in at the time of the survey. These data were not analysed because of the scale of the effort needed and the need to bring closure to the empirical phase of the research.

Question 5 was an open question.

A copy of the questionnaire is included in the Appendix (pp. 248-249).

3.7.2.3 Studying patterns of maintainership

The term 'maintainership' is used hereafter because of the importance of the role of maintainer, the rights and responsibilities, as indicated in this section, which in F/OS involve a broader and more critical set of issues than passive maintenance of code implied by the description 'maintenance'.

Examination of maintainership patterns introduced three problems. First, the boundaries of GNOME and KDE needed to be defined. They are both huge projects involving a large number of spin-off applications utilizing different aspects of the development platform. Second, the notion of maintainer needed to be defined. Third, in order to identify them, a protocol had to be developed.

In choosing which modules to include in the research, it was decided first that only those included in the official project release would be examined and second that an effort would be made to maintain a balance between modules. This was deemed necessary in order to be able to draw comparisons between the two projects.

The study of the GNOME project involved examination of 110 modules included under Platform and Desktop Sources that were shipped with the 2.10 release. The KDE project involved 191 modules included under KDE Libraries, KDE base, KDE multimedia, KDE PIM, KDEdevelop, KDE Network, KDE webdevelop, KDE utils, KDEdsdk, and Kde accessibility, shipped with the 3.4 release. A detailed list of all the modules included in the study can be found in Appendix (pp. 250-251). The difference in the number of modules arose from the fact that the two projects organize their components differently. The rules that were developed to set project boundaries mean that QTtoolkit, an integral component of the KDE development platform could not be included in the analysis since it was not included in the KDE release and is developed externally to the community. The GTK toolkit, however, was included in the GNOME study, since it is included in the release.

In order to maintain a basis for comparing the two projects KDE Edu, a part of KDE that aims to develop applications relevant to educational activities, was not included in the study, since GNOME does not have an equivalent project. KOffice was excluded for the same reasons.

The role of maintainer involves certain rights and responsibilities and is connected with notions of authority and ownership. Usually the first group of project maintainers are those that initiate a project, and individuals that subsequently become maintainers have often been making major contributions and are among the most knowledgeable committers. In many cases there is no official hand-over of maintainership and the role is often assumed informally by what are then considered 'default maintainers'. As indicated by the analysis of the quantitative data in Chapter 6, it is also quite common for a project to have more than one maintainer.

A maintainer is responsible for answering queries and responding to problem reports. In the case of large projects he/she is the contact person for coordinating with the individual or the team in charge of the release. Maintainers' release duties usually include providing an updated report about the bugs solved since the last release cycle and making certain that their module integrates well with other parts of the code-base.

Maintainers usually influence the technical direction of their projects, providing the vision behind their development. This is associated with their right to peer review, approve and incorporate patches into the code base. Even where an individual has CVS commit rights[31] it is generally considered bad form to forgo the maintainer's approval. Lastly, it is usually the maintainer that grants to a contributor CVS write access (or supports an application for it).

Similarly to Foundation and e.V. members, the examination of maintainers focused on experienced and committed contributors. Unlike the Foundation and KDE e.V., however, where non-coders can participate, the maintainer networks consist only of programmers and are therefore more technical in character.

To identify KDE and GNOME module and project maintainers different strategies were pursued. In the case of GNOME the maintainers' names are usually included in a text file titled 'MAINTAINER(S)' in the release tarballs.[32] This list was cross-referenced with information on the relevant websites. The affiliations of the 42 maintainers who were not among the GNOME Foundation respondents were identified from the following information sources:

[31] This means that a person has already been sanctioned to commit changes to the code base.

[32] A tar archive, or tarball in the jargon, is a group of files compressed together as one, which is used to package source code distributions.

- CVS commit mailing lists. The information sought here was whether an email address was a corporate one.
- Blogs/websites. These usually include references to professional lives and frequently include CVs.
- GUADEC schedules. Maintainers are often invited to speak at developer conferences. Conference schedules usually include a small bio outlining contribution and employment status.

The names and affiliation of GNOME maintainers were validated by two long-term community contributors.

In the case of KDE, very few tarballs included updated 'MAINTAINER(S)' files. To overcome this the bugs databases' maintainers were contacted; they compiled lists of individuals who had 'closed' the most bugs. 'Closing' bugs involves providing patches or solutions for faults in programs. This is can be done either alone, that is, by writing the solution and commiting it, or by committing on behalf of someone else. Consistent with the definition of maintainer, the names on the compiled lists were of those considered to be 'default' maintainers. The protocol used to identify the affiliations of 82 maintainers that were not among the KDE e.V. respondents was the same as followed for GNOME. The names and affiliations of KDE maintainers were validated by a long-term KDE contributor.

The effectiveness of the protocols developed to identify maintainers and their affiliations was confirmed by the fact that very few changes were made by the developers who validated the produced data.

3.8 Analysis of quantitative data

As indicated at the beginning of this section, the aim of the study was to highlight substantive patterns of contribution and maintainership among the examined groups of contributors. To that end, a detailed meta-coding scheme was developed that categorized the reported areas of contribution in five major areas of development:

- **Core/Platform Modules:** This code includes critical elements of the GNOME and KDE development platform and their core technologies included in the releases.
- **Main Desktop Modules:** This code includes the core components of the GNOME and KDE desktops, such as the File and Window Manager, the Multimedia Framework and the principal Email and Personal Information Managers.

- **Secondary Desktop Modules**: This code includes modules that are secondary to the function of the desktop, such as user applications or secondary libraries.
- **Development Tools and Processes:** This code includes tools and activities related to unspecified fixes, development of tools for developers, integration/porting and some non-coding activities that are integral to development, such as coordination of releases.
- **Peripheral Activities:** This code includes non-coding activities, such as translation, documentation, conference organization, etc.

The first two areas include the more critical aspects of the GNOME and KDE development platform and desktops. Contribution to these parts of the code base usually signifies a high-level of expertise. Table 3-4 [33], presents in detail what each code includes in GNOME and KDE.The same coding scheme, with the exception of 'Peripheral Activities' was employed to codify patterns of maintainership.

These codes were developed after consulting the following sources:

- for GNOME the website http://developer.gnome.org/arch/ and the 'Overview of the Gnome Platform' written by Shaun McCance http://developer.gnome.org/doc/guides/platform-overview//.
- for KDE the overview at http://developer.kde.org/documentation/library/kdeqt/kde3arch/ and various other online sources on the KDE website like the API docs http://developer.kde.org/documentation/library/3.5-api/kdelibs-apidocs/.

The scheme was validated by two GNOME and two KDE long-term contributors. As the quantitative data analysis in Chapter 6 indicates this method of coding was successful in revealing some qualitative differences in patterns of contribution and maintainership across the examined groups of participants.

[33] Table 3-4 does not list all the examined modules. Its aim is to provide a more detailed view of how the coding scheme was implemented.

Phase 3		
Area of Contribution/Maintainership	**GNOME**	**KDE**
1 Core/Platform Modules: Development Environment: Graphical Interfaces Component Model Core Technologies/Libraries: Virtual File System, Configuration and Lock-down, Printing/Imaging, Accessibility Libraries Internationalization framework	GTK+, GDK, Pango, Glib, Glade Vfs, GConf, atk, atspi, Gail, Intltool,libgnomeprint, libgnomeprintui, libart, Gnome Canvas, Bonobo, Orbit2, libIDL, libxml and libxslt, D-Bus, Cairo, Libgnome Libgnomeui, Esound, Scrollkeeper, Gnome icon theme	Kparts, KHTML, KIO network transparency Most of the projects included in KDE Libs: KDE Core, KIO,Kiosk, Internationalization: Qt & kdecore/klocale & kdesdk/kbabel, KJS: KDE Javascript Library
2 Main Desktop Modules: *Core Desktop components*: Desktop Icons & File Manager, Application and File Registries, Window Manager, Control Centre, Session Management, Storing passwords, Display Manager Email/Group/Personal Information Management Clients Multimedia Framework	Gnome Panel, Metacity, Control Centre, GNOME keyring Manager, Gnome menus, Address Book and Calendar (Evolution), Gnome desktop, Gnome menus Control Centre, GDM-Gnome display manager gconf-editor, gnome terminal, gnome volume manager gst plugins, Gnome applets,Libwnck, Libxklavier	Most of the projects included in KDE base that are considered to belong to KDE proper and in particular: Konqueror-libkonq KWIN, Kcontrol KSMserver, KDEnetwork KDM-KDE display manager Kttsd: accessibility, Ksysguard
3 Secondary Desktop Modules: Secondary desktop components Secondary libraries User applications	Eog, Galeon,Gnopernicus. gnome speech, gnome net status, gnome net tool, gnome speech, gnome system monitor, gnome system tools, Gnome utils, GOK gpdf, gtk-engines, gtkhtml, libgsf, libgtop,librsvg start-up notification, nautilus cd burner, nautilus media soundjuicer, system tools backends, totem vino, vte, ximian connector, zenity	Kwalletmanager, knots,kpm ktimemon, ktop, kwrite, Quanta Kommander, kmousetool, kmagnifier, kmouth
4 Development, Tools & Processes: Bug triaging, Development tools, General (unspecified) Bug fixes, Language/platform bindings, Integration work & porting, Packaging, Release planning		
5 Peripheral Activities: Translation/internationalization, Documentation, Usability,Website maintenance.Accounts management, Conference organization, Themes design, Background design, Icons & Fonts design		

Table 3-4: Breakdown of GNOME and KDE modules

3.9 Conclusion

This chapter indicated how the research questions underlying the investigation of power relations in F/OS communities which were presented in Chapters 1 and 2 are operationalized and outlined this study's research design. The research strategy relies on a three-step research design that is based on a combination of qualitative and quantitative methods. The interviews conducted during the first phase of the research provided a first-level view of various technical and social aspects of the projects and highlighted the richness of the discourses underlining F/OS development, and the complexity of the structuring practices employed to organize access and control participation. The nested case studies functioned as foci of meaning that consolidated the interview findings and elaborated the themes of commercialization and peripheral participation. The quantitative analysis tested certain hypotheses that emerged from the qualitative phases of the study in order to map some of the structural relations and interdependencies that formed between paid and volunteer contributors, and core and peripheral community. The limitations of the adopted research design, including those imposed by the characteristics of the selected primary case studies and the applied analytical methods, are discussed in detail in Chapter 8, section 8.3.

The methodology which has been outlined introduces significant innovative elements into the examination of the dynamics of participation and collaboration in F/OS projects. These are the methodological broadening of the framework of investigation by considering F/OS communities as constellations of practice; the iterative, three-step research design, where the insights gained through each stage inform the focus of the next; and the methods and protocols developed for the quantitative examination of structural aspects of participation.

4 Communities and the economy: commercialization of mature F/OS projects

4.1 *Chapter overview*

This chapter begins to untangle some of the threads of power that weave together relations and values underlying participation and development in F/OS by presenting the findings related to the dynamics of commercialization in F/OS communities. The chapter draws from data gathered from interviews, ethnographic observations and online material to highlight the complex interdependencies that develop between communities and corporate actors. [34]While taking into account the institutional arrangements mediating their relations, this investigation takes a step further and examines the implications of corporate involvement at the level of routine development.

The chapter is organized as follows. Section 4.2 'F/OS Communities in/and the Economy' develops a preliminary framework for examining the commercialization of F/OS by reviewing contributions made on the subject. Section 4.3 presents findings related to paid developers, one of the most important ways for corporate actors to become involved in community development. Section 4.4 focuses on the Gstreamer/Fluendo nested case study and examines some other aspects of commercialization, namely the tensions between innovation and maintenance and the way volunteers and paid developers manage boundaries between companies and communities. Section 4.5 examines the challenges and successes of the GNOME Bounty Hunt which represents a new way for corporate actors to incorporate their agendas in the community development process. Section 4.6 summarizes and draws implications from the main findings of the study of relations of power underlying F/OS development.

4.2 *F/OS communities and the economy*

This section outlines a preliminary framework for examining the issue of commercialization of F/OS and describes how the empirical data represents some of the complex relationships entailed in this process. F/OS software communities have often been considered to occupy a space between the organized supply of products and services offered by firms and the wider, emergent, market dynamics of software

[34] For a detailed breakdown of volunteer and paid developers were interviewed in phase 1 see Table 3.2. For a detailed view of the conferences and meetings attended see section 3.2. The GNOME Bounty Hunt nested case study (Phase 2) comprised interviews and the Gstreamer/Fluendo SL case (Phase 2) consisted of six. For the method of analysis and the coding scheme developed to analyse the interview findings (in Phases 1 & 2) relating to commercialization, see section 3.6.

production, supply and demand. O'Mahony (2002), for example, refers to community managed open source projects as a new type of commercial actor and von Hippel (2002) considers F/OS development as the template for an innovation process which is primarily driven by software users.

At the same time it has often been argued, by both practitioners and academics, that the motivations of developers contributing to F/OS projects are significantly different from those underlying proprietary software development (Elliot and Scacchi, 2003b; Lakhani and Wolf, 2005; Raymond, 2001a). This view has been interpreted as suggesting a contrast between two kinds of economic rationality; the gift economy and the exchange economy. Whereas the first, based on the principle of reciprocity, upholds the idea of software as a public good and is often associated with abundance, the second relies predominantly on monetary flows and market transactions and is organized around a scarcity of resources. As we saw in Chapter 1, this dichotomy forms an integral part of accounts that emphasize the importance of the values of the hacker culture in the success of the movement.

This idea is being challenged on several fronts by research drawing attention to the interconnections between the two spheres of economic activity. David Lancashire (2001), for example, employs a political economy perspective to explain the distributions of F/OS development worldwide. When recalculated on the basis of Internet usage in each country the data indicate that F/OS development activity is fairly low level in the United States, intensifies in Northern European countries and is fairly high in Southern Europe. His conclusion, which is consistent with Lerner and Tirole's (2002a) arguments that F/OS development provides opportunities for developers to tap into global corporate networks, is that participation in F/OS is more intense in countries with relatively low levels of ICT investment and growth, where it is seen, especially by young developers, as a means to attain higher wages. Brief reference to the role of paid developers in GNOME development is made by German (2002). Combined with the accounts of large scale studies of F/OS development communities (David, et al., 2003; Ghosh, et al., 2002; Lakhani and Wolf, 2002) which indicate that projects are driven by the contributions of seasoned ICT professionals, these findings illustrate some of the possible connections and overlaps that exist between the exchange and the gift economies.

The desire to understand the motives, the dynamics and the principles of F/OS development does not simply stem from intellectual curiosity. The economic significance of F/OS communities both in terms of their output and as a repository of highly skilled and potentially free labour force is very appealing both to commercial and public actors.[35]

[35] The enrolment of F/OS development communities forms an important part of many regional and national industrial policies and constitutes an integral element of many corporate strategies aiming at competitive advantage in the software market.

The literature focusing on this subject is relatively underdeveloped and largely falls into two categories: those that consider issues of commercialization from the perspective of volunteer communities and those that focus on issues of F/OS appropriation from the perspective of companies. The most notable examples of the first category are O'Mahony's (2002) study on the new kinds of organizational structures communities develop in order to accommodate companies' needs, while safeguarding their work, and Freeman and Siltala's (2004) study on the hybrid practices developing within GNOME and Open Office due to corporate involvement.

Their approach is counterbalanced by Lin's (2005) examination of the practices of developers in open source firms that partake in F/OS development and the resulting forms and tensions from this collaboration. Other examples of the second group of studies include research into companies' motivations for becoming involved in F/OS development (Mustonen, 2005), accounts of business models and strategies aiming to appropriate returns from F/OS (Behlendorf, 1999; Dahlander, 2004; West, 2003), and a study of the types of relationships that develop between firms and F/OS communities (Dahlander and Magnusson, 2005). O'Mahony and West (2004) have elaborated a framework that takes into account the dynamics of collaboration in commercially initiated F/OS projects. The study of the commercial appropriation of F/OS is taken one step further by Brian Fitzgerald (2006), who in his article 'The transformation of open source software', examines the more mainstream and commercial instantiations of F/OS which he terms OSS 2.0.

This chapter complements these contributions by highlighting issues of commercialization and commodification from the point of view of F/OS communities. More specifically, it draws attention to the way corporate involvement affects the development of projects and the dynamics of cooperation at the meso and micro levels. In doing so it provides an emergent view of the complex interdependencies that develop between corporate and community players and begins to untangle the meshwork of relations that sustains and in turn is sustained by the idealized and routine practices and discourses that underlie participation and development in F/OS.

4.3 *Paid developers in community managed F/OS projects*

Companies contribute to projects in many ways. They give money to support promotional activities and community conferences including venue and travel costs, and provide hardware. The website and the CVS tree of the GNOME project, for example, is hosted by Redhat Inc. and KDE's is hosted by SUSE/Novel Inc., with Trolltech ASA providing general DNS (Domain Name System) facilities. However, the most important, and many would argue the most potent form of involvement of companies in projects is through contributing labour, by committing a number of their employees to the development process.

This section highlights the different forms of employment arrangements and sponsorship of employed programmers and draws attention to the different kinds of relations and tensions that each implies.[36] There are, for instance, considerable differences, in terms of acceptance by the community, between paid hackers who previously worked as volunteers, and employed contributors external to the community. This typology aims to form a basis of understanding of the different kinds of dependencies that form between communities and companies. For example, companies' strategies employed to recruit hackers who are already on 'a critical path'[37] in the community development process mean that in most cases it is the programmers contributing in key areas of the project (core developers) who are hired. This may have significant implications for the development of the project.

4.3.1 Types of employment and sponsorship

From the data gathered from the interviews it seems that, depending on their particular mandate, paid developers fall into the following categories:

Those that have a free sponsorship. This group does not receive any clear instruction from their employers about what they should be working on. They are usually volunteers who are more or less expected to work in the same areas they were contributing to before they were hired. An example here is a Czech KDE developer who is working for Novell Inc./Suse Labs.

Those that have a clear mandate from their employers in terms of what they do. Some developers, for example, working in Linux distribution companies, are expected to help integrate different aspects of the project into company products or build on top of their projects' platforms to create commercial applications (the case of Fluendo SL).

Those that have 'KDE or GNOME-friendly' jobs. These are people, usually hired by companies or departments of companies, that have a strong F/OS orientation. They are usually hackers who are expected to work developing proprietary company products, but who are also allowed to spend part of their time working on community projects. This type of agreement, can be formal or informal, and resembles a part-time free sponsorship.

Those that are hired or compensated to solve a particular problem or develop a specific application. This type of involvement is akin to sub-contracting. Sun Microsystems Inc, a company that is heavily involved in GNOME, for example, hired developers from Wipro Ltd, a large Indian software house, to help on certain aspects of GNOME that their own GNOME team did not have time to deal with.

[36] The definition of 'paid' adopted here is different from the one given by Lakhani and Wolf (2005), which includes programmers who contribute to F/OS during working hours without the sanction of their employer.

[37] I would like to thank Luis Villa for suggesting this term.

4.3.2 Working from the outside: external contributors and community development

In addition to the previous categorization, there is an important distinction to be made between contributors that previously worked for the project on a volunteer basis and external programmers who were commissioned to work on it but had no previous involvement in the community. Proprietary developers that are brought in to work on community projects have to learn the ways of the community and adjust to the rhythms and the demands of F/OS development. 12 interviewees thought that this group has significant difficulties in adjusting to the often unstructured work flows of the community. This is how a GNOME contributor described an early encounter between GNOME hackers and company employees:

> T.: Oh, yes, I do remember one meeting with Sun in the early days, just an accidental one, cause some GNOME hackers were visiting Dublin and we went up to the big Sun headquarters there and we said "release on time? No, we never release on time!" and the look of the people who were actually working for a living on it at Sum was that of absolute horror! "What do you mean you don't release on time, we are relying on that date to ship to our customers!
>
> (Sean[38], 09/03/04, p.11)

Some of these issues have been resolved. The regularization of the release cycles of GNOME and the KDE, for example, allows companies to time the release of their own products better. Although 11 interviewees stressed the frequent incommensurability of community and corporate agendas and processes (often described as a divide between open source and proprietary development) eight, in fact, believed that, at the same time, communities are doing more to accommodate companies' needs. Despite the progress being made, however, there appear to be more gaps than areas of contact in the more formal aspects of community-corporate cooperation.

Sometimes companies' Quality Assurance (QA) processes are incompatible with those of the community or corporate legal requirements hinder considerably the integration of the latest code release in corporate products and services.[39] On the community's part companies' contributions to the code base can either be too specialized for the overall aims of the project and, consequently, the issue of 'pushing the patches upstream' (that is, integrating them into the main development tree) does not make any sense or they are relevant. In the latter case companies usually pursue integration because it saves on the time and effort needed to maintain the changes made to the code base. The ideal scenario is the maintainer of the relevant module

[38] In order to preserve anonymity interviewees have been given aliases. The date refers to the date on which the interview was conducted and the page number to the page of the interview transcript.

[39] When Nokia, for example, adopted the Gstreamer multimedia framework for their 770 Internet Tablet the time-consuming process of reviewing the legal framework resulted in the Nokia development team adopting an earlier and hence 'buggier' version of Gstreamer.

reviews the proposed changes and if they are found to be satisfactory, they are adopted. Sometimes the community is quite reluctant to accept such contributions, especially if they imply significant changes to the overall structure of the module.

An underlying assumption in the acceptance of code submitted by paid contributors is that it is done on the same kind of meritocratic basis that applies to the rest of the community. This is how Al framed it:

> If, in a project like GNOME people are very afraid, well not afraid, but aware of not letting the project like GNOME or KDE, or the Linux kernel or something like that be completely hijacked by commercial interests, like they, and I keep saying 'they', but I mean 'we' really, I am part of this, we are very happy to have commercial interests contributing resources by companies like Sun and Redhat and Novell and all of these big companies, like it is fantastic that they are devoting essentially hundreds of thousands of dollars of developer resources to help us out. But at the same time it has to be done on the same basis as when I contribute something to GNOME. It is, I am putting my code out there, it is open source, the community accepts it or rejects it, and there is no special favours given to someone just because they come from a company with a big name and sort of millions of dollars in revenue each year.
>
> (Al, 24/06/04, p.7)

Some of the issues that Al raises are aptly illustrated by the case of Wipro Ltd (see section 3.5.1, Chapter 3). Wipro Ltd is an Indian Software House which was subcontracted by Sun Microsystems Inc. to work on the GNOME project. Although they were expected to work with the community, Wipro Ltd developers were not expected to join it. Their success was measured according to specific metrics (bugs fixed, bugs logged). Due, however, to their lack of prior knowledge about the GNOME code base and the inadequacy of documentation they had to resort to asking for information on community mailing lists and IRC channels. The presence of a dozen developers asking questions all at once stressed the learning resources of the community and created a stir among volunteer developers. A further complication was that their contribution was not visible to the community. This is what Martin, a GNOME developer who was working at the time for Sun Microsystems Inc, observed:

> So they were on IRC asking us very intelligent questions but you would see very little end result for the questions, you know what I mean? So that kind of thing invisible, it is impossible to see what they are actually doing, or how they are actually achieving anything. Now, behind the scenes I know exactly what they are doing, what they were doing, they have been working very, very hard for Sun, but in terms of being visible, kind of within the community?
>
> (Martin, 11/07/04, p.21)

In addition to the idea that corporate involvement is based on the same criteria as individual volunteer contributions, 11 interviewees acknowledged the importance of the resources companies brought to the project but did not consider them irreplaceable. This is how Gerri responded when asked about the Sun's involvement in GNOME's documentation:

> Interviewer: Do you think that this might be a problem in the future, because it sort of indicates might be relying more and more for its documentation, you know, on a company?
>
> Gerri: Well, I don't think that it will be really a problem, because if the company keeps being involved, it is not a problem at all, but if suddenly the company leaves, I think that there would be some people who will replace the people who are leaving. So I don't see that as a big problem, but, well, I don't know if this would be the case, but, for example, I feel that the contributions of Sun's people, well, there are less contributions than, say, two years ago. I am not sure but I feel that and I do not think that this had an impact on the project.
>
> (Gerri, 18/10/04, p.14-15)

This view of companies as external actors in F/OS projects, upon which the creation of organizational bodies such as the GNOME Foundation and KDE e.V. assigned with the task of mediating relations between the community and companies is founded, is dramatically different if the focus is on the role of employed community members.

4.3.3 Working from the inside: employed community members and project development

> Well, nearly everybody at Ximian originally were sort of young hackers plucked from IRC and given a job
>
> (Sean, 09/03/04, p. 12)

Developers who previously worked as volunteers, carry with them a network of connections and extensive know-how of community processes that both facilitate their work in terms of its acceptance by the community and help connect corporate and community teams.

In companies such as Sun Microsystems Inc. that have large teams of developers working on F/OS projects, such people sometimes assume the role of intermediaries between the rest of the corporate team and the community.

> Leroy :Oh, I interact with the community very regularly, but I would have to say that I am not as involved as many people on my team, I try to play somewhat of a different role. I tend to be the person on the team who interacts with other people, other teams within [name of the company] to make sure that the needs of other teams are addressed by [KDE/GNOME],
>
> Interviewer: Ah, interesting.
>
> Leroy: Like, for example, I spend a lot of time interacting with the [name] team and they are a low bandwidth product and in order for GNOME to work well on [name of the commercial application] it has to work well within a low bandwidth environment. And, so they get very interested in fixing certain bugs which are not probably that interesting to most [KDE/GNOME] developers, who run Linux on a stand alone desktop, which is a very different environment. So I probably spend more time interacting with other people in [name of the company to find out what [KDE/GNOME] needs to do to meet their need and then I work with the community to get those needs addressed. But it is kind of a little hard sometimes, because the people in the [KDE/GNOME] community

> aren't necessary interested in these issues that are only really a problem for an internal group to {name of the company}.
>
> (Leroy, 18/03/04, p. 2)

Company strategy to recruit volunteers who are already on a critical path in the project, who are heavily involved in central aspects of development and who have demonstrated that they are highly competent technically, means that very often employed contributors are maintainers of key parts of the project.[40] In fact, ten interviewees suggested that paid developers are often involved in central aspects of the project. It is plausible that their potential as volunteer contributors combined with the opportunity to work full-time in the project enhances their position as core developers, as central actors who not only have the ability to control key aspects of the project but who also have an interest in keeping an eye on its overall development. This is what Al said when asked who are the people that try to keep abreast of all the developments in the project:

> Mm, well very much, like putting them in broad categories, the sort of people who do tend to have an across the board view are, anyone on the release team to a certain extent, I don't know if you know, but there is a release team who is responsible for getting the product out every six months or so. Now obviously they have to keep an eye across the board on what's going on. And also the guys again, who are working for the commercial companies, the RedHats and the Novells and the Suns to have a sort of direct interest, in whatever happens in GNOME, they are going to have to eventually make it into their particular branded version of GNOME, like whatever goes out in RedHat Linux or in Fedora, or whatever goes out in the old Ximian desktop or in Sun's Java desktop system, these are the guys who sort of have, they kind of have a vested interest in the implications and again, because of the fact that they kind of doing it or also because they are good GNOME community members they tend to try to resolve these problems, think about resolving these problems in a good way, but they tend to be the people, like if I sort of think who are the people who, if I had to go to someone and say, look, who is completely across GNOME, it's these people who hold those positions in companies, who sort of have this kind of interest to stay involved out of more or less self-defense otherwise at some point they are going to be completely overwhelmed by trying to understand the whole thing, so its better to keep up to date.
>
> (Al 24/06/04 p. 10-11)

At the same time employed community members demonstrate an increased sensitivity to community issues that makes them cautious with regard to balancing community and company interests. Although some paid developers admitted that on certain occasions managing community-company boundaries was challenging, 12 interviewees did not think that the roles of paid developer and community member was inherently incompatible. In fact nine paid contributors[41] suggested that, they considered that their work did not, necessarily, contradict with their communal identity. An important factor that affects these developers' relations with their

[40] The quantitative analysis of patterns of contribution and maintenance among GNOME Foundation and KDE e.V. members and among the two projects' maintainers presented in Chapter 6 helps to build a more precise picture.

[41] This includes the interviewed Fluendo employees.

employer and influences their everyday working life, concerns the links that their employers have with the projects and with the F/OS community in general. There appear to be at least three different cases here. KDE and GNOME paid contributors are employed by large, established companies such as Sun Microsystems Inc, F/OS companies, predominantly Linux distributors such as RedHat Inc. Novell/Suse Inc., and Mandrake, or companies such as Ximian (which in 2003 was bought by Novell Inc.) that have been created as direct spin-offs of community projects. Needless to say, the links are not always straightforward. Klarälvdalens Datakonsult AB is a Swedish company created by one of the founders of the KDE project, Matthias Kalle Dalheimer. This small company develops product tools for developers and provides training materials and courses for developing using Qt, a toolkit developed by Trolltech, which is employed in the development of KDE. Almost all its employees worked for a number of years as volunteers on the KDE project and have transferred the patterns and tools and cooperation that they developed there to the new corporate environment. This is what Sebastian had to say:

> Well the funny thing is that we apply the same way of working...the KDE way of working will apply in our company as well. We use the same tools, since we are all working from home, we have the same communication mechanisms, we use Kmail and IRC and it works very well, like KDE itself works.
>
> (Sebastian, 23/08/04, p.13)

It is interesting to note that in the case of companies with strong ties with the F/OS world the confluence between community and corporate interests appears less complicated. The fact that they do not have to try to explain community processes and policies makes their relationship with management easier. More importantly, in contrast to working in a company with a less developed F/OS culture, face-work in terms of upholding community values and ideals seems to be conducted not in opposition (us and the rest of the company) but collectively (we in free/open source).

However, as the next section shows, even in this idealized scenario many problems remain unresolved and resurface both at the individual and the collective levels.

The case of community integrated employees suggests that the issue of community-corporate relations is more complex than the view of the company as an external actor in the development process mediated through appropriately developed institutional interfaces suggests. More specifically, it appears that in most cases the connections between companies and projects develop at different levels of involvement and hinge on complex interpersonal dynamics. The majority of elected members on the GNOME Foundation and the KDE boards are either hired or self-employed contributors. One contributor in fact suggested that bodies such as the GNOME Foundation offer provide an opportunity for companies involved in projects to keep an eye on each other and avoid nullifying each others' actions.

The Gstreamer/Fluendo nested case study allows us to examine the way community and corporate boundaries are managed and highlights a different set of issues related to the commercialization of F/OS communities, such as the tension between the demand for stability and the desire for experimentation.

4.4 *Gstreamer and Fluendo SL: At the intersection of company and community*

Gstreamer (http://gstreamer.freedesktop.org/) is a F/OS multimedia application framework loosely connected to the GNOME project since it shares many of its programming libraries. The Fluendo SL (http://www.Fluendo SL.com/) company was founded in January 2004 by Julien Moutte and Pascal Pegaz and when the study was conducted in 2005 it employed six full-time developers, most of whom had been working on Gstreamer as volunteers, including a programmer who was considered for a long time to be the maintainer of the project. The company's flagship product at the time was Flumotion, a streaming server that utilizes Gstreamer.

This section takes a snapshot of a particular time in the cooperation between the Gstreamer community and the Fluendo SL company, a period at the beginning of 2005 when volunteer and paid developers tried to resolve the issue of the project having forked into two competing branches and coordinating Gstreamer's next major release. It draws on six interviews conducted during this period with Fluendo SL developers and Gstreamer volunteers and on ethnographic observations during the Gstreamer/Fluendo SL summit. This section elaborates some of the issues presented above and, at the same time, highlights a new set of interdependencies forming between companies and communities, such as when the company has to accommodate the learning curve of the community or when corporate involvement becomes a sign of a project's maturity.

4.4.1 *The Gstreamer/Fluendo summit*

Between 17 and 21 February 2005 at Fluendo SL's offices in Barcelona Gstreamer and Fluendo SL developers came together in order to solve some technical issues and plan the project's next major release. In particular the summit aimed to create a consensus between the company and the volunteers on the schedule and the priorities for Gstreamer's next (0.10) release and to find a solution to the development of two separate project branches which represented two different design approaches in terms of how the framework handled the scheduling of tasks (threaded vs. event driven). The first branch was led and continuously improved upon by one of Fluendo SL's employees who had been the lead developer in Gstreamer and the second, more experimental one, was initiated by two volunteer developers.

The fact that the first branch was technically more complete, since it had been developed on a full-time basis, meant that in terms of proof of concept Fluendo SL's threaded approach had a considerable advantage over the more incomplete and hence more difficult to test event-based approach. Despite this, Fluendo SL's developers were open to the possibility of accommodating the more experimental approach even though in the interviews they acknowledged that it represented a direction that they would not have pursued if the community had not shown an interest in it. After three days of intensive talks it was agreed that the next release would be based on Fluendo SL's branch, but would incorporate some elements from the event driven approach.

Scheduling the next release proved to be slightly more challenging. Fluendo SL programmers were sensitive to the fact that the incorporation of their leading developer's changes to the project's main branch needed to be done gradually so that the community would have time to assimilate them. On the other hand, the company needed the next stable release in order to sell the new versions of its products. When discussing the list of changes that needed to be incorporated in version 0.10 a volunteer programmer remarked that it could not be ready before Christmas 2005. The agreed release date as set out in a posting on planet Gstreamer and the Gstreamer mailing list was May 2005. Version 0.10 was finally released on 05/12/05; in other words, the volunteer programmer's estimate was the more accurate one. However, 0.10 did not ultimately implement ideas that had started to be developed in the event-based branch. In the meantime one of the two developers of the event-based branch decided to withdraw from the project. The next section links the case of Fluendo SL and Gstreamer to wider questions about the balance between the corporate and the communal.

4.4.2 Between stability and experimentation: F/OS projects as products and research projects

The alternation between stable and more experimental releases is a well-established practice in the F/OS process and is reflected in the conventions used for numbering the different versions of projects.[42] The balance between experimentation and refinement is defined by the way release cycles are organized. In KDE and in GNOME the release schedule is organized on the basis of different 'freezes', that is, periods during which coders are free to modify certain aspects of the project but not

[42] In F/OS projects versions usually consist of three integers, separated by periods, i.e., 3.4.1. The first digit is the major number and the second is the minor. The last indicates the patch level. Thus, 3.4.0 is the initial release of the 3.4 version of the program, with the next upgrade being 3.4.1. The second digit is also important in that it determines whether or not the program is a development or stable version. If it is an odd number this indicates that it is in development. Both the stable and development versions will have releases at various intervals. Once 3.4 goes into a feature freeze, it will be branched out into 3.5. Though minor development (bug fixes, minor updates to drivers) will continue, major reconstruction will be going on in 3.5. Once 3.5 has reached a release state, it will be released as 3.6.0 and the process will resume.

allowed to work on others, so that the other groups of contributors, such as translators and documenters can catch up.[43] During freeze periods developers are supposed to concentrate on fixing faults and bugs in their programs, a task that is considered by many to be less exciting than the process of adding new features. When talking about the different project tasks the need to undertake some of the more mundane aspects of development, such as bug-tracking or writing documentation, was frequently acknowledged. For example, Gerri a volunteer coder pointed out that what makes F/OS projects work is that they are "pretty well organized in terms of people doing what they want to do and people doing what they should do" (Gerri, 18/10/04 p. 2). When asked whether development is always fun in F/OS another contributor replied that: "it can't be fun, unless sometimes it isn't fun, otherwise it is always the same (Clint 19/02/05, p. 7). Finding the right balance between the period when developers can be at their most creative and freezes is not a straightforward matter. As Roger indicated having too many or too lengthy freezes can push developers away:

> Actually, the KDE 3.0, I think it was, was released very strictly, more in a GNOME sense to set, to say 8 months before we release in December and the freeze was very early, so that nothing happens afterwards because <<name>> who did the release, the manager, his job before, is what I call a control freak. Because he can't live with not knowing everything and he basically lost himself within this. So, and because he had put such a big, strict control over the whole release schedule, there were many people who were distracted and looked for other projects, where they can develop while the are waiting for the freeze to be over, so this is the need you need, you lose. Because, if they once, if they find a project that they are more interested in, they are lost as free developers.
>
> (Roger, 22/08/4 p.13-14)

As pointed out at the beginning of this chapter, however, the involvement of companies creates a more pressing demand for release cycles to become more standardized and for projects to be better maintained. The tensions that the change from a feature based release, which basically means that the code is released whenever all the planned features are incorporated into a time based release and where the major releases are made within a specified period, have been documented by O'Mahony (2002). At the same time, interview findings indicated that corporate involvement in projects was acknowledged to improve their stability giving them a more polished and professional finish. Because of this corporate involvement is often taken as a sign of project maturity.

This was also one of the expectations of Fluendo SL's development in Gstreamer. All six interviewees expected that Fluendo SL's involvement would greatly improve Gstreamer's code base. Victor, a volunteer hacker was, however, concerned that this would limit their freedom to experiment:

> The product probably even is more polished, at least I think it has been an important part of GNOME that the people have, the companies have made

[43] This will be examined in greater detail in Chapter 5.

> people fix all those bugs that come up there, so GNOME really doesn't crush anymore. And it was bad in the early days. On the other hand, I often fear that companies reduce innovation, because the developer, well two things really, the developer develops on the stuff he did for fun before now 8 hours a day, so after those eight hours, he doesn't want anymore, though after those eight hours he could really could what he wants and not company stuff.
>
> (Victor, 18/02/05 p. 9)

Roger, a paid KDE developer seems to agree that the involvement of companies in F/OS can overturn the balance between innovation and maintenance:

> So I for one, or <<name>>, or <<name>>, are paid for doing KDE development. We will always do this. But we are not as creative usually. So <<name>> fixes bugs, whatever comes as feedback for the Window Manager, or <<name>> has some, some, kiosk mode implemented for administrators. We will always refine this more and more, but we are not coming up with brand new ideas. The brand new ideas always come from people that come from outside that say, well I used your Kmail and I think I've got some input that I've seen in an application and I know how to code, here is the patch and then you look at this and say, yeah, a very nice idea, the implentation lacks a bit, please use here and here this different method, because you can't know everything and usually can't find the perfect solution for a problem, that you want. [....].So, and this, you can only have this new contributors if you are open for new stuff that you can't think of beforehand, if you are open, if you are frozen, if you do not control for a long time over the year.
>
> (Roger, 22/08/4 p. 13-14)

There is an alternative view. In some instances it was argued that many innovative things have originated from the work of paid developers, people who have the time to become deeply involved in more aspects of development than the average volunteer and who usually, as indicated by Victor, continue to hack on projects outside their working hours. Although an assessment of the sources of innovation in F/OS projects is beyond the scope of this thesis, it is clear that companies and paid developers are primarily considered as stabilizing and normalizing factors in the F/OS development process. This seems to have an ambiguous effect on community development. On the one hand, the commodification of F/OS projects appears to bring a degree of structure that helps organize participation, but on the other it is seen as having the potential of imposing an instrumental view on the development process that may hinder the influx of experimental ideas and new contributors. The next section throws a different light on the balance between corporate and community agendas by examining the way they are handled by employed community members.

4.4.3 *Managing boundaries and roles: At the intersection of community and company*

According to the three Fluendo SL employees who were interviewed the transition from volunteer to full-time employed contributor was not difficult. This was largely due to the familiarity of their employer with community requirements and the F/OS process of development. According to all three interviewees working in physical proximity with the other members of the community that were hired brought, more context to their interactions, allowing them to get to know each other better and achieve more in terms of cooperation. At the same time the opportunity to work full-time on a project that they previously could only afford to hack on in their free-time made it possible for them to "get more out of it, get more verifiable results" (Michael, 19/02/05, p. 1). As we have seen Gstreamer's lead developer, who before joining the Fluendo SL team had disappeared from active project development for about a year, managed to re-establish his threaded approach once he started working for the company. When asked whether it was difficult to balance corporate and community requirements the three interviewees suggested that at that particular time Fluendo SL's goals for Gstreamer coincided with community aims. They did not deny, however, that this could change in the future. Supporting DRM (Digital Rights Management) in Gstreamer was a choice that the Fluendo SL team expected would raise some issues within the larger F/OS community. They tried to counterbalance this controversial choice by supporting the Xinth Foundation, an organization that promotes open media formats. In terms of practical, day-to-day, considerations what the Fluendo SL developers found challenging was to make the community realize the faster rhythms of development and to replicate online the discussions that they had amongst themselves to enable the community to follow and participate in developments.

However, even in the cases of companies with strong ties with the F/OS world the convergence of community and corporate agendas is less straightforward than is frequently acknowledged. I was present at a discussion between a developer who had been working for a long time in a company that was created as a GNOME spin-off and a Gstreamer contributor who had been recently hired by Fluendo SL. For reasons of confidentiality I will refer to the first as Dan and the second as Jim. In discussing their plans for their new company Dan advised Jim to think through to the boundaries ("lines") that they would need to put in place between the community and the company ("corporate"). He mentioned the case of some developers, who started a company that was created as spin-off of the F/OS project "who were thinking community, community, community and then one day they woke up and they realized, that, shit! I am corporate". The earlier these lines are drawn according to Dan, the less trouble Jim's people would have in negotiating their double identity. "What are you going to do?" asked Dan "if your clients push you to deliver a product?". At the same time, in large information technology (IT) companies with weaker ties with the F/OS world, as a GNOME employee that had worked for Sun Microsystems Inc. suggested, sometimes employed hackers who found that an assigned task undermined the F/OS agenda, would "drop things on the side" (Martin, 11/07/2004 p.7) and let somebody else pick it up.

In the Gstreamer community the blurring of the line between community and corporate was sometimes deemed confusing. This was to some extent due to Fluendo

SL's policy of occasionally subcontracting parts of the development to volunteer hackers. A volunteer interviewee also suggested that the fact that the Fluendo SL developers were inevitably the most cohesive and the most productive group in community development skewed the volunteer character of the project. However, one of the most interesting implications of corporate involvement in community projects was pointed out by a Fluendo SL hacker who observed that:

> yeah, but I want to say is actually, you have the free time developers that sort of say, ok, we trust this knowledge to that company more or less and you, like in the case of Ximian, if, if Evolution for example, lots of programmers there working on the mail clients, if Evolution goes down, what happens to it? I mean I'm an open source programmer, I am contributing to it, but, I mean have a lot of trust put into this monolithic, that monolithic company that is doing this thing and that is providing me support, what happens if they go away? Do the developers there think, well, isn't that dangerous this company thing there. I have no idea. [...] I don't know, I mean I've never worked for an open source project where a large chunk of the code or a large chunk of the knowledge was with some entity that could go away, I wouldn't know how would that be. Because you see, you see actually with QT that the Trolltech is like controlling the GUI toolkit and well what happens if they go down, what happens there? Most of their, there is not a lot of outside contribution to that system, but it is a core part of KDE. They are putting quite a lot of trust into that company, which is, which is one reason why GTK for example is a community project where it's like managed by the community not by this single entity
>
> (Michael, 19/02/05 p. 11)

When asked to think about the implications of sponsoring companies withdrawing from F/OS projects, five interviewees believed that the community would eventually be able to pick up the threads of development and continue the work, albeit at a slower pace. The alternative scenario suggested by Michael is interesting because it draws attention to the investment of development knowledge in people and organizations which is inevitably lost when they decide to withdraw from community development. This belief is supported by the redundancy of labour and effort often associated with F/OS. However, there is evidence that sometimes the process is not so simple. When the maintainer of GNOME's window manager, Swordfish, was hired by Apple, it was impossible to find somebody to take over the project. This was because the program was written in a special version of Lisp that no one was very familiar with. After having to deal for almost six months with a program that remained unmaintained, GNOME developers eventually switched to a different manager called Metacity which was written in C, the language of the entire platform.

The relation between Fluendo SL and Gstreamer shows us that although in the cases of companies with strong ties with the F/OS world the convergence of corporate and community agendas may appear less problematic than in the case of larger companies with less established connections, there are still several issues resulting from the permeating corporate-community boundaries. First, the accelerated rhythm of development tests the community's ability to follow developments. Secondly, despite the influx of new resources, the blurring of the boundaries between the corporate and the communal seems to work in favour of companies more than the communities. Lastly, the investment of knowledge in corporate entities and paid employees raises significant questions about the future of the projects to which they contribute. The next section examines the GNOME Bounty Hunt which represents a new way for communities to interface with external actors.

4.5 *The GNOME Bounty Hunt: testing a new interface between companies and communities*

The GNOME Bounty Hunt was a contest that was initiated by Novell Inc. at the end of 2003. It consisted of a series of monetary awards ranging from $15 to $2,500 for improvements to a series of GNOME and GNOME related programs with the overall aim of better integration of different aspects of the desktop. The contest was quite controversial, partly because it institutionalized monetary rewards in a space conceived to be primarily driven by intrinsic motivations, and partly because of the particular way in which it was handled by Novell Inc. This section examines the main issues and challenges that Bounty Hunt gave rise to on the side of its organizers, the participants and the volunteer community. This examination is pertinent to our investigation of the commercialization of mature F/OS projects as bounty programs are becoming increasingly popular in the F/OS world, providing a way for companies and other actors, such as public institutions, to interface with the communities other than participating directly in the development process (Krishmamurti and Tripathi, 2005). Google, for example, in 2004 introduced the 'Summer of Code', an initiative that shared $2 million across a large number of F/OS projects distributed as bounties.[44] Section 4.5.1 provides the backdrop to the contest and section 4.5.2 focuses on the main issues that arose, such as the complications that it created in the established development process.

4.5.1 *Background, history and rules of the GNOME bounty hunt*

The GNOME Desktop integration Bounty Hunt was conducted in two phases over a period of roughly 18 months. The initial announcement was made on the GNOME Footnotes website on 22 November 2003. The contest consisted of 45 bounties ranging from $15 to 2,500, with the average bounty prize being $540 and the most frequent reward $500. Overall Novell Inc. planned to distribute $24,315. Many of the bounties related to the integration of data from the calendar, mailer and address

[44] This initiative is still running (May 2007) and has awarded cash prizes to students who successfully completed a task, termed a 'project', associated with F/OS. Google sponsored 419 projects in 2005 and 630 in 2006. The students had to be supported by an organization that was required to provide them with a mentor. These organizations could be established F/OS communities, education institutions or businesses engaged in F/OS development. The students who completed the project to the satisfaction of their organization received $4,500. For more details see http://code.google.com/summerofcode05.html and http://code.google.com/summerofcode.html (last accessed 10/2006).

book, with a substantial number of bounties relating to Evolution, the email program developed by Novell Inc. The second phase was announced on 4 August 2004. It consisted of the 34 bounties remaining from the first phase of the contest. At the end of the second phase 21 bounties were claimed which in total amounted to $12,200.

The idea of such contests is not new in the F/OS world. Krishnamurti and Tripathi (2005) provide a classification of various bounty contests based on their goals (software development, maintenance, security, specific bugs, documentation), source of funds, formality and bounty amount while providing some illustrative examples of different types of contests. Freeman and Siltala (2004) provide an overview of the main issues involved in the GNOME Bounty Hunt in their article 'Freedom and profit: how suits and hackers are working it out on the desktop'. According to Nat Friedman, one of the founders of the GNOME project, Novell Inc.'s VP, who came up with the idea of the contest, the Bounty Hunt had three aims:

- To urge people to think about the integration of personal information on the desktop.
- To bring new developers in to GNOME.
- To urge the community to start thinking about bounties.

In order for a bounty to be claimed, the work had to be accepted by the maintainer of the relevant module and become part of GNOME's subsequent release. In addition to maintainers patches were reviewed by a panel of experts. Although the contest rules did not specify who has the final word, the maintainer or the panel of experts, it appears that the panel had the power to award a bounty even if the patch was accepted by the maintainer, which is what happened in two cases. Participants were also urged to announce their intention of working on a bounty in Bugzilla, the GNOME project's system for tracking bugs. The biggest change in the rules, which was made in the second phase, was a clause prohibiting participation in the contest of Novell employees working in the Ximian division. This was precipitated by two cases, which eventually were eliminated however.

4.5.2 *Successes, challenges & the future of bounties*

The GNOME Bounty contest gives rise to a series of questions regarding how bounties are defined, managed and integrated into a project's development process. At the same time the contest is connected to some wider concerns regarding the balance between public and private interests, and extrinsic and intrinsic motivations in F/OS. This section provides an overview of these issues as they emerged through the interviews and the analysis of the relevant public online material and begins to consolidate and connect some of the themes related to commercialization to that of peripheral participation, providing the background to Chapter 5. This analysis is organized on the basis of the contest's three primary aims.

Urging people to think about the integration of personal information on the desktop: specifying, implementing and integrating bounties

In contrast to the Google bounties that were agreed upon through consultation with the project maintainers, the GNOME Desktop Integration bounties were solely defined by Novell Inc. Although this was generally acceptable, since the bounties were sponsored by them, there were certain objections with regard to the specific technical choices the bounties represented, which were seen as undermining the power of maintainers and the agreed upon policies of the project. This applied both to certain higher-level choices, such as the controversy over the decision to implement certain features in FireFox rather than in Epiphany, GNOME's default Internet browser, as well to specific requested features that maintainers may not have necessarily found suitable for their modules. At the same time, the decision to connect acceptance of bounties to the community process created significant difficulties for maintainers who found themselves under pressure to review and integrate patches often just before major releases, the most intensive periods in the project's life cycle. These two points were also highlighted by Krishnamurti and Tripathi (2005). Another element that substantially added to the maintainers' workload was the fact that several bounties were not clearly specified in terms of how they should be implemented. As a consequence some of the submitted patches did not conform to the architectural requirements of the module which, in turn, required significant additional work on the part of both the maintainer and the contestant.

Many of these issues were expressed in the public mailing lists as part of an overarching concern regarding the hosting of the contest on the GNOME community website. Many developers believed that there should have been a clearer demarcation of the contest as a Novell Inc. and not as a community initiative. The controversy escalated when a Novell Inc. employee announced the second round of the contest as an initiative of the GNOME Foundation, which administered the awarding of bounties on behalf of Novell Inc.

Bringing in new developers in GNOME: adding, cannibalizing and reprioritizing resources

In order to better understand the agenda underlying the contest it is necessary to point out that the bounties were also meant to kick start community involvement in Evolution, a company initiated program where the size of the code base and the intensive rate of development had created a substantial barrier to entry for new volunteer developers (Tonni, 30/05/05). As we have seen this was a danger Fluendo SL employees were very aware of and strove to avoid. However, despite individual assessments that the bounty was indeed successful in attracting new developers, there is a lack of clear evidence as to: a) whether the contest managed to attract developers completely unrelated to GNOME development or whether it simply redistributed labour by refocusing priorities within the project; and b) whether the new, external developers stayed on as volunteers.

In fact the question of who the bounties appeal to appears to be more complicated than Krishnamurti and Tripathi's (2005) analysis suggests. They proposed that participants in bounty contests are likely to be:

> "Developers who do not have other opportunities in a recession. Low-end developers with low skill levels. Student developers who wish to learn to prepare themselves for market positions. Developers in countries where wage rates are low."
>
> (Krishnamurti and Tripathi, 2005: 16).

However the two contest participants who were interviewed had experience in the modules that the bounties applied to. One of them had in fact been the maintainer of the relevant module. Furthermore, as we have seen, two of the bounties were solved by Novell Inc. employees. Lastly, as a maintainer pointed out on the GNOME Foundation's mailing list, it is possible that some of the participants would have participated in GNOME regardless of the monetary reward. Although this evidence is clearly limited, it suggests that concerns regarding the bounty's redistribution of labour within the community are legitimate and that contestants may not necessarily fall within the categories proposed by Krishnamurti and Tripathi.

It is worth noting that an aspect of the contest that greatly appealed to contestants was the fact that the bounties represented a clearly articulated task list. As will be shown in Chapter 5 this is a major issue for new developers who often do not know where to start and need assistance in choosing an appropriate task.

Urging the community to start thinking about bounties: the balance between the private and the volunteer

The contest was largely set up as an experiment. Nat Friedman viewed this as part of an effort to create a general bounty system in F/OS, a view that he also expressed in his blog. Friedman regarded bounties as an opportunity to bring users closer to developers. He conceived a system where not only companies, but individual users could collectively bid to improve certain aspects of programs. One interviewee in fact suggested that bounties might be a way to put some pressure on maintainers to implement certain features which they might not have considered otherwise:

> I think demand will drive direction, so, if you were to have this general bounty system like where anybody could think of an idea I think sometimes people will think of things that don't drive whatever the maintainer wants and that you know, but if those things are well funded enough and enough people get behind then just going to and the maintainer will just to have to change his opinion or have his project aborted.
>
> (Harry, 31/05/05 p. 6)

This point becomes particularly significant if we consider it from the perspective of those who conceive F/OS development as driven primarily by users (von Hippel, 2002). Bounties, therefore, raise some important questions about how requests are expressed, prioritized and implemented, and by whom, issues that touch upon the balance of power between individuals controlling the development from within the community, namely maintainers, and external and peripheral actors, such as companies and users. This distinction which underlies the construction of F/OS as a pure, volunteer space fails to take into account the degree to which projects are already commercialized through the presence, for example, of community members on the company payroll. This point was made clear by several by the interviewees, including Peter, a participant in the contest:

> "A lot of people I know think that open source should definitely be something that people work on for free, although there are a lot of hired open source developers at RedHat and Novell, so I don't know why that should be the case. There is a lot of controversy regarding how the bounties were initially set up. Those projects didn't even know that their programs were going to be a part of these bounties, so I paid attention to some of that."
>
> (Peter, 10/06/05, p. 2-3)

What seemed to fuel the debate, however, was not the question of whether commercial interests were already part of the process of development, but the degree to which they were acknowledged and acted upon. It was suggested that there is a delicate balance between intrinsic and extrinsic motivations that bounties and intensive corporate involvement have the capacity to overthrow. Preserving this balance is as much to the benefit of companies, which would otherwise not have the multiplier effect of volunteer labour, as to the wider community. Victor a volunteer Gstreamer hacker expressed his views on the matter very eloquently in his blog:

> 7 April 2004
>
> Luis from Ximian made me now write something about the issue I talked about with him last Guadec and that still gets worse: Companies vs Communities.
>
> I still have a bad feeling in my stomach about the GNOME people getting too corporate and starting to a) raise the barrier to entry into the GNOME hacking world, b) hiding decisions from the community and c) steering GNOME into corporate waters far too much. [...] But the most important thing for me is that the people I interact with are more and more employees that need to get a job done, not people that hack on their project for fun. (Add the bounties here if you want.) IRC nicks end with @redhat.com or similar instead of .edu, .pl or .net. And this is especially bad because the newbie help is missing. Because helping newbies offers no short-term benefits and people nowadays need to get a job done. But that reduces new blood coming into the GNOME world. And I believe that without new people joining, GNOME will not be able to scale. GNOME needs all those testing volunteers that run broken cvs builds and report bugs, categorize the bugs, do translations, create new artwork, provide patches or write new tools. You'd need to throw a lot of money in that direction if you wanted to pay for that. And it is my strong believe that you can only attract people to your cause, if you provide a community to attract the people. If someone is interested in GNOME but doesn't know how to get involved, joins #GNOME and speaks a shy "hi" and then noone answers, GNOME might

> have just lost a valuable contributor. I still know this day three years ago when I first joined #gstreamer and said a shy "hi" and people did talk to me. I liked it more than the other places I tried, so I stayed.
>
> You can say that in business terms, too: If no one had taken the 15 minutes to convince me to stay, GNOME would not have gotten contributions for free that would probably have cost a 6-digit Dollar-amount to get professionally."
>
> (http://www.advogato.org/person/company/diary.html?start=14http://www.advogato.org/person/company/diary.html?start=14 , last accessed 15/01/06)

The additional complication that bounties bring to the debate regarding volunteer and corporate agendas is that they institute a separate and, therefore, more visible process for coordinating external and peripheral involvement. Whereas many paid developers are well integrated into the community process, making it quite difficult to distinguish their roles as community members from their roles as employees, bounties are clearly demarcated as bids, parts of a tendering process, that are the hallmark of the exchange economy. What distinguishes bounties from the involvement of external developers is that they can be taken up by volunteer community members, thus making it more difficult to identify who ultimately drives development.

4.6 Conclusion

This chapter has examined the dynamics of commercialization in mature F/OS communities by developing a typology of engagement of companies and communities based on the involvement and the role of paid developers and the new kinds of interfaces between them, namely bounty hunt contests. The analysis highlighted two distinctive aspects of corporate involvement in F/OS communities. The first is based on the view of corporate players as actors external to the community. This perspective emphasizes the resources that companies bring into projects, especially with regard to the more mundane aspects of development, such as documentation and usability, and points to the way communities respond in order to accommodate corporate needs, such as through the regularization of their development cycle. This view is underlined by the idea that when companies pull out of a thriving project, the community will eventually be able to pick up the threads of development and continue the work, albeit at a slower pace, a belief that is supported by the redundancy of labour often associated with the gift economy. This aspect of corporate involvement is also consistent with the view adopted in studies examining the new types of organizations, such as the GNOME Foundation, the GNOME Advisory Board and KDE e.V. that communities create in order to protect their work and regulate their relations. The participation of externally paid developers and, in particular, the acceptance of their code in the main development tree, is expected to be judged on the same kind of meritocratic basis that applies to volunteers. Another expectation regarding corporate involvement concerns the transparency of company contributions. Maintainers want to be aware of the work being done by paid developers in order to avoid duplications of effort and prevent technical clashes.

The second aspect of corporate involvement in F/OS is based on the examination of the role of community integrated paid contributors. Companies often recruit volunteers who are on a critical path in the development process, individuals that contribute to critical aspects of development and have a proven high technical competence. At the same time some external developers can also become community members through a gradual process. This perspective undermines the idea of corporate involvement as a factor entirely external to the community and highlights the interdependencies that develop between companies and communities at the level of interpersonal relations. Although their number is limited, community integrated employed developers often assume a leading role in projects. They are often the maintainers of core modules and they keep abreast of developments that affect the whole of the project. At the same time they demonstrate an increased sensitivity to community issues that makes them cautious with regard to balancing community and company interests. Their close community ties in turn endow them with a legitimacy that cannot be easily questioned, since it appears to rely upon relations of trust that have developed over time.

The analysis has indicated that an important factor that affects these developers' relations with their employer and influences their everyday working lives, concerns the links that their employers have with the projects and with the F/OS community in general. Developers employed by large corporations sometimes think of themselves as belonging to a separate community within the company, whereas developers who are employed by F/OS companies conceive of their working relations as an extension of their identity as community members.

The case of Gstreamer/Fluendo, however, revealed that even in the case of companies with strong ties with the F/OS world the convergence of community and corporate agendas is not as straightforward as it might first appear. The accelerated rate of development made possible through the work of paid developers often raises the barrier to participation by volunteer programmers. In addition, the investment of knowledge in corporate entities and paid employees raises significant questions about the future of the projects to which they contribute. In some cases the issue of picking up where a company has left off is much more complicated than is frequently acknowledged as it often involves a radical change of direction. Moreover, the blurring of the boundaries between the corporate and the communal, as the case of Fluendo SL indicates, seems to work in favour of companies more than communities. As the analysis indicates companies and paid developers are primarily considered as stabilizing and normalizing factors in the F/OS development process. This seems to have an ambiguous effect on community development. On the one hand, the commodification of F/OS projects appears to bring a degree of structure that helps organize participation, but, on the other, it is seen as having the potential to impose an instrumental view on the development process that could hinder the influx of experimental ideas and new contributors.

While communities set up interfaces between themselves and the market in order to protect their autonomy and their work, corporate actors make the forces of the market visible through initiatives such as bounty hunt contests. What seems to fuel the debate regarding how bounties are defined and managed, however, is not the question of whether or not commercial interests are part of the process of development, but the degree to which they are acknowledged and acted upon. Many respondents believed that preserving this balance of power in favour of the gift economy is as much to the benefit of companies, which would otherwise not have the multiplier effect of free, volunteer labour, as to the wider community which wants to maintain its autonomy and safeguard its future.

While premised on this idea of abundance, the gift economy binds individuals together through a web of interpersonal relations that is essential for carrying out their work. In F/OS it appears that the economic is more deeply embedded in the social than in many other expressions of economic life. This characteristic of the gift economy, which often results in making the economic value of labour difficult to quantify, is reified by the shared discourses of community and sociality that often underlie what we recognize as the hacker culture. In turn, this discourse forms an essential part of face-work, of the public upholding of the shared community values that are so necessary in maintaining the community's identity and which are instrumental in attracting new volunteers. The case of the GNOME Bounty Hunt showed us how the discourse of the market, the exchange economy, can be mobilized to connect the community back to its users. This case study indicates one of the possible blind spots of the gift economy. When relations of production, that is, work relations, are so closely coupled with social relations it may be difficult for outsiders to have a serious role and a say in the development process. This problem is amplified by the meritocratic basis of F/OS. In Chapter 5 the way the growth of F/OS projects and the need for skill sets other than coding are considered in light of the idea of meritocracy as technocracy.

Figure 4-1 summarizes this chapter's key themes and highlights their interconnections.

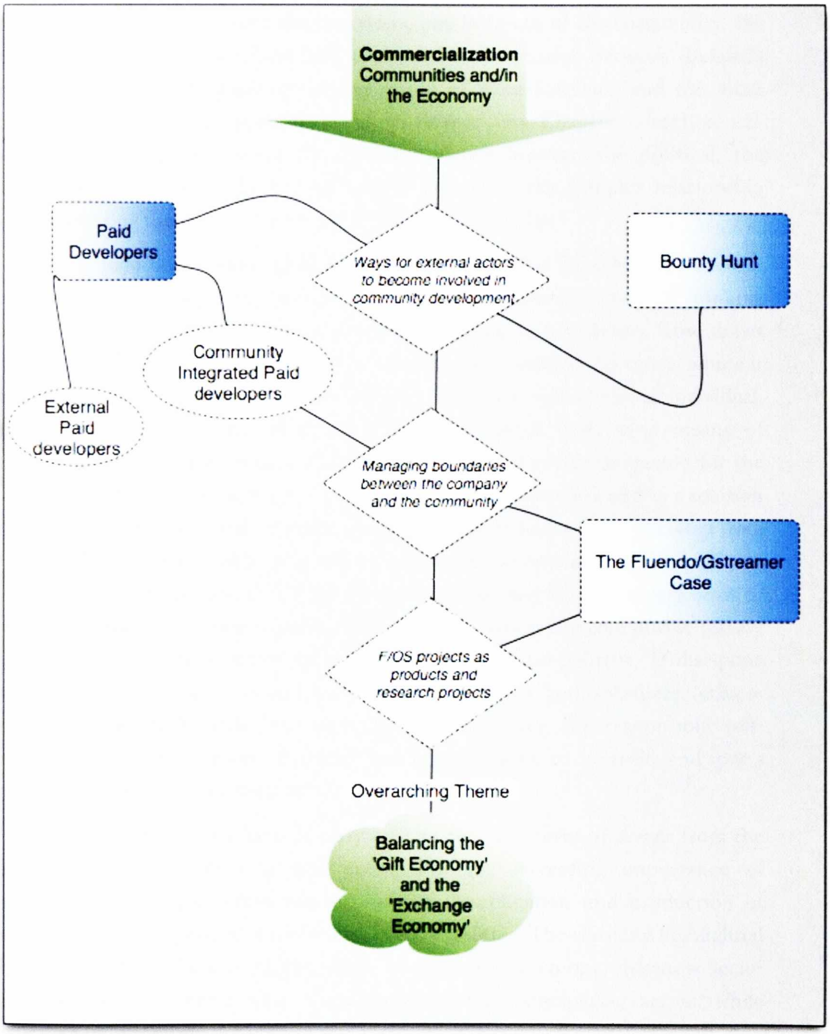

Commercialization Communities and/in the Economy
Paid Developers
Ways for external actors to become involved in community development
Bounty Hunt
Community Integrated Paid developers
External Paid developers
Managing boundaries between the company and the community
The Fluendo/Gstreamer Case
F/OS projects as products and research projects
Overarching Theme
Balancing the 'Gift Economy' and the 'Exchange Economy'

Figure 4-1: Commercialization key research themes and their interconnections

As indicated, although the economic significance of F/OS not only in terms of the value of the products, but more importantly in terms of the volunteer labour, forms part of the shared understanding among contributors, not everyone agrees where the line between the gift economy and the exchange economy should be drawn. Historically, differences between the two views, one in favour of the community, the other in favour of the market, have been expressed as the division between Richard's Stallman more idealistic, 'philosophical' approach of 'Free Software' and the more practical, market oriented approach of 'Open Source' (see Chapter 1, section 1.2). However, both camps recognize the interdependence between the political, the economic and the social. Is there a way then to talk about this complex relationship without reiterating the division between community and market?

In his 1999 book *Powers of Freedom: Reframing Political Thought* Nikolas Rose adopts Foucault's idea of governmentality to suggest some alternative ways of thinking about contemporary regimes of power and their histories. In his analysis Rose draws attention to the pervasiveness of the idea of community in political discourse, where it has come to signify a 'third space' between the state and the 'rights bearing' individual. According to Rose the idea of community has emerged both as a means of problematization and a means of a solution. As a means of problematization for the receding role of the state in many aspects of social and economic life and as a solution in "its promise of a new moral contract, a new partnership between an enabling state and responsible citizens based upon the natural bonds of community" (Rose, 1999: 186) . Rose goes on to suggest that this concept of community lies at the very heart of a new power game in the same way that the Panopticon was at the core of disciplinary power. This power of community operates in a field of ethico-politics: "If discipline individualizes and normalizes, and bio-power collectivizes and socializes, ethico-politics concerns itself with the self-techniques necessary for responsible self-government and the relations between one's obligations to oneself and one's obligations to others" (Rose, 1999: 188).

F/OS appears to be a kind of carry over of this new form of power from the political field to the economy and exemplifies the increasing importance of communities as strategic actors not just in the dissemination and production of knowledge, but in the production of valuable commodities. The tensions highlighted in this chapter are indicative of the effort to appropriate, co-opt, this new socio-technical space, to inscribe it to the agendas of the participating actors, while safeguarding its autonomy.

The view of F/OS community projects that emerges through these findings is one of a complex space that is organized recursively by a multitude of informal and formal practices, strategies and discourses and by networks of people and relations that divide and integrate different aspects of development and practice. This chapter has untangled some of the threads that weave together relations and values, pervaded by the sometimes converging, sometimes oppositional discourses of the gift economy and the exchange economy, of the community and the market, of the social, the political and the economic. The analysis suggests a framework for understanding F/OS as an expression of a new form of power. Chapter 5 pursues a different thread of relational power and highlights F/OS communities as constellations of practice. This shifts the focus of the investigation from the interdependencies forming between communities and external actors to the dynamics of cooperation developing between different groups of contributors.

5 Peripheral participation, learning and the division of labour in F/OS communities

5.1 Chapter overview

This chapter continues the investigation of the different power relations underlying F/OS development by presenting the findings related to peripheral participation. In addition to highlighting issues of access and the role of learning in structuring participation, this chapter draws attention to a somewhat understudied area concerning the groups of contributors that supplement and support code development. This aspect of peripherality we term 'autonomous peripherality'. The characteristics of non-coding teams and their relations with programmers is becoming an increasingly important aspect of F/OS development.[45]

The chapter is structured as follows. Section 5.2 Between communities and practices: peripherality, learning and power in F/OS projects develops two aspects of peripherality, legitimate peripheral participation and autonomous peripherality, and situates them within the context of other studies of F/OS. Section 5.3 presents the findings related to legitimate peripheral participation, first (section 5.3.1) from the perspective of novice developers (newbies) and then from the point of view of senior developers (5.3.2). Section 5.4 outlines the findings concerning autonomous peripheral participation. Following a summary of the characteristics and organization of some of the non-coding teams (section 5.4.1), the focus is put on how their work is coordinated with the main developer community (5.4.2). Section 5.5 situates autonomous peripherality in the context of how projects evolve, and presents an analysis of two initiatives aimed at improving its impact and organization, the KDE Quality teams and the Open Usability project. Section 5.6 draws together the main points of the analysis and discusses their implications for understanding the dynamics of cooperation in F/OS.

5.2 Between communities and practices: peripherality, learning and power in F/OS projects

In Chapter 2 a framework for the investigation of F/OS projects as communities of practice was established by building on studies adopting this perspective within F/OS and in the field of organizational learning. The conceptual framework for this study is characterized by the examination of projects as constellations of practice rather than uniform spaces built around the same practice, and by the emphasis placed on power and access, two aspects of Wenger's and Lave's theory that have remained largely unexamined. This section situates the two types of peripherality explored in this chapter, intercoder peripheral participation and autonomous peripherality in the context of contributions related to learning and peripheral participation in F/OS. The first type of peripherality concerns the process of the legitimate peripheral participation within the group of programmers. The second type of peripherality relates to the participation of non-coders and highlights their distinctive characteristics and relations with the community of programmers. These two expressions of peripherality are examined within the context of a conceptual framework that draws attention to how learning practices and relations between and across different groups of contributors are embedded in relations of work and production.

Learning is not only featured as one of the main motives for participating in F/OS, but learning practices and processes, such as peer-review, are often regarded as constitutive elements of the F/OS development model. Studies related to learning in F/OS can be organized into two broad groups. The first group consists of studies that examine the role of tools and the technical characteristics of projects in the learning process. The second group includes studies that focus primarily on issues of socialization and joining. A good example of studies belonging to the first category is Shaikh and Cornford's (2004) examination of Version Control or Concurrent Version Tools (VCT or CVS). They indicate that CVS are both an important medium (at the level of code, at the level of the community) and a product of learning (since their design embodies lessons learned at the organizational level). In the same vein Baldwin and Clark (2004) draw attention to the importance of code architecture in organizing and inviting participation and argue that the modularity of the code is a critical factor in the decision to participate or not in a project.

An example of the second group of studies is Ducheneaut (2005), which adopts a social-technical approach to the study of how relationships of new F/OS developers develop dynamically with the social and material aspects of projects. His analysis indicates that: "successful participants progressively construct identities as software craftsmen, and that this [learning] process is punctuated by specific rites of passage" (Ducheneaut, 2005: 323). The question of socialization and integration of newcomers was also the main focus of von Krogh et al.'s (2003) paper on 'Community, joining and specialization in open source software development'. These authors argue that newcomers who eventually received a CVS account adopt specific joining scripts, that is, patterns of behaviour involving levels and types of activity required to become a community member. The notion of 'joining script(s)' will be addressed later in this chapter. Both these studies adopt an oversocialized view of learning that is consistent with functionalist accounts of communities of practice, predicated upon the idea that communities are built upon consensus, shared values and continuity.

In Chapter 2 attention was drawn to the limits of the idea of F/OS communities as homogenous CoP. It was pointed out that although F/OS communities are recognized as being intensively specialized (von Krogh, et al., 2003), the only group other than coders mentioned in the literature is users. The primary role of users in the F/OS development process is providing feedback, for example, by indicating flaws (bugs) in programs or suggesting new features. Weber (2000) suggested that even the most inactive users, who are frequently perceived as free-riders, play an instrumental role in dissemination of the software. The issue of peripherality as a related but potentially distinctive sphere of activity from that of the main community has also gone largely unexamined in the literature. Although some aspects of peripheral participation, such as lurking (Musamichi, et al., 2003; Preece, et al., 2004), have been investigated, the only study that has directly addressed the question of peripheral members appears to be Zhang and Storck (2001). Zhang and Storck examined a travel forum hosted by a Chinese service information company and defined peripheral members on the basis of their posted messages. Specifically, a peripheral member is defined as someone who posted fewer than 30 messages. "Under this criterion 987 out of 1065 who ever posted a message during the sampling period were classified as peripheral member" (Zhang and Storck, 20005: 5). Zhang and Storck argue that peripheral members contribute in aggregate to information exchange as much as non-peripheral members. In a study on the role of peripheral developers in F/OS, Laknani (2006) pointed out that peripheral members, which in this case are defined as developers without commit rights, played an important role in the development of new features and collective problem solving.

This chapter complements these contributions by highlighting aspects of joining and integration of programmers within a project's dominant community of practice, and examines the framework of their collaboration with non-coding communities that take an active part in developments. First, the distinguishing characteristic of the present contribution lies in the examination of learning and joining under the demands of continuous production. Second, it extends the notion of peripherality to take account of the need for labour to be divided and coordinated in growing F/OS communities involving more than one type of contributor. Both aspects of peripherality, therefore, are defined within this research qualitatively rather than quantitatively.

5.3 Intercoder peripherality: new developers (newbies) learning and participation

This section focuses on the first aspect of peripheral participation which concerns the integration of new developers as their skills develop and they gradually move from the periphery of the project to its centre. The section highlights learning and integration from two different perspectives. Section 5.3.1 outlines the findings from the perspective of newbies; it examines some of the main barriers to participation and highlights specific patterns of joining. Section 5.3.2 examines questions of integration from the perspective of experienced developers, contributors at the centre of the project, and draws attention to the tactics and rules of thumb they use to assess the potential of new developers and decide whether or not to assist them. It is argued that understanding learning in F/OS needs to take account of both these perspectives, not only because experienced developers constitute important information and access points, but because their viewpoint highlights the extent to which learning processes are scaffolded by the requirements of continuous production.

5.3.1 Learning and contributing: the newbies' perspective

Despite their open collaborative character, F/OS communities are characterized by significant barriers to entry. The stumbling blocks to participation can be categorized into three broad categories. First, there are difficulties associated with the technical aspects and tools of F/OS development, such as the use of CVS. Secondly, there are conceptual difficulties related to understanding the development process and the architecture of the program, how they are set up, how things fit and how they are expected to be put together. Thirdly, there are difficulties related to how newbies situate themselves in the development process, where they start and what are the tasks most appropriate to their skill levels.

Even before they reach the point of fiddling with the code, new developers have to learn how to download (check-out), build and install the program's sources. This allows them to run the latest, in-production, version of the code, a prerequisite for participating in the ongoing development process. Installing this development snapshot is far from straightforward. Once a newbie overcomes this initial hurdle and writes a patch, it is necessary to resolve the question of how to submit it in the correct format. This involves writing a set of rules that will allow people to build the submitted code along with the rest of the sources. Both these processes, checking out and checking in code, involve not only a degree of familiarity with the CVS tools but also a conceptual understanding of how "things are put together" (Dean, 22/03/04 pp. 2-3). Moreover, the incorporation into the main development tree of a patch depends not only on its technical merit, but also on its conformance to the maintainer's view of what are appropriate features and its compliance with the architecture of the module. The analysis of the GNOME Bounty Hunt highlighted this difficulty (see chapter 4).

Although documentation may provide information on some of these issues, such as how to use the CVS, solutions to the more conceptual problems involved in development are often not provided. Moreover, even if such information exists and even if it is updated, finding the relevant documents can often be an arduous task. 11 interviewees pointed out that most common problem in relation to learning concerns the fragmented character of the documentation and other online resources that were available. This is how one interviewee described it:

> Yeah, I think if you, to get involved into the thing is really hard I think, because basically everything is very disparate and you have the bug database here, you have the CVS access there, and there is no central documentation for how to get involved into all this and you have to learn by yourself all the different entry points, you have to learn that you have to read the mailing lists, that you have to read the CVS commit to stay aware of what is happening, you have to read different websites, but, all these information sources are set up by people who know each other, so every time something new happens, everybody in their little circle is aware of it. But for outsiders it is very hard to stay, to stay in front of everything, you have to keep looking everywhere and notice new stuff and just go there and take a look. It has been better recently, because of all the websites and Planet GNOME is very useful for that, because it centralizes the whole development information, but you still have to read everything and take out what is really important and throw out all the personal stuff, so, I think that it is a lot of work, but at the same time, it is really interesting, so it is not really that bad...
>
> (Neal, 19/10/04 pp. 5-6)

The intertwining of the social and technical aspects of information in blog syndication sites, such as Planet GNOME, described by Neal, which is a characteristic of CoP, may account for the frequently employed practice of lurking on the project's development mailing lists. The role of lurking in the socialization of new developers has been highlighted in other studies. Von Krogh et al. (2003) quantified how long developers remain at the periphery, which was defined by them as the period during which emails were posted but no active contribution was made, and concluded that new contributors post an average of 23 emails before being given a CVS account. The public, archival character of mailing lists and their use as repositories of knowledge makes the posting of a message a non-trivial affair, especially for new developers. As section 5.3.2 shows, experienced developers usually assign considerable importance to how newbies comport themselves on mailing lists, especially the ones that bear the most important development traffic. The more formal character of the development lists is counterbalanced by the more informal and fleeting discussions taking place on a project's IRC. This medium is considered the best way for newbies to receive immediate support and to get a feel for the development as it evolves in real time.

In addition to building a sense of community, developing an understanding of a project's processes and getting to know who is responsible for what, lurking on the mailing lists and getting on the IRC channels appears to provide one of the most effective ways for new developers to find a task that is suited to their interests and skills. Eight interviewees considered finding a suitable task to be one of the major steps on their way to becoming recognized contributors. It is interesting to note that three developers initially became involved in F/OS developments by assuming peripheral tasks, such translating, or by concentrating on self-contained development tasks such as bug triaging.[46] In certain cases, as in the case of Flint, a newbie coder, having tasks or projects explicitly addressed to new developers appears to greatly

[46] Triaging is a QA process that involves confirming good and reproducible bug reports from the projects' bug tracking tools, to establish exactly which actions generate faults in the program. Bug triaging is conducted not only to resolve invalid and not reproducible bug reports, but also in order to prioritize bugs, and to find and resolve duplicate bug reports.

facilitate participation, not least because the people who initiate them usually take on the role of mentor. All interviewed newbies placed a high value on mentoring relationships not only because of the learning they facilitate, but also in terms of the reassurance they provide that their efforts are appreciated. Although responsiveness is often deemed to be one of the key characteristics of F/OS, getting the right people to pay attention to one's suggestions and work is not always straightforward.

Despite these difficulties, and in some cases because of them, interviewees described their experience of participation and collaboration in F/OS as an educational, sustained learning experience.

Thus, the newbies interviewed for this study describe integration as a slow learning process during which they build up skill sets and community knowledge and position themselves in developments by choosing suitable tasks. This process is often understood as a sustained learning experience that often complements formal education. The views of experienced developers regarding newbies on this process were rather different

5.3.2 Learning and the imperative of production: the developers' perspective

> There is a certain kind of make up that it takes to become an open source developer and I think that the process of becoming one can of to a certain extent ensure that that's true.
>
> (Sam, 14/04/04, p. 14)

Whereas newbies describe the process of integration in the development community as predominantly a learning process, the discourse of more experienced developers regarding new contributors is more production-oriented. This shapes their expectations with regard to the behaviour and performance of newbies and influences decisions to help them.

One of the characteristics most valued in new contributors and F/OS development in general is self-reliance. In fact, nine developers referred to it as one of the most important characteristics of new developers. The ability to independently navigate the maze of F/OS projects and find answers is connected to two main factors in F/OS development. The first is the availability of the source code, which is considered to be the ultimate necessary documentation; access to the source code means that potentially the answers to most questions are available. The second is the limited time senior developers in an intensive production oriented environment can afford to spend teaching others. The rapid release rate of F/OS developments and the fact that many developers work on a volunteer basis makes time a very scarce resource in F/OS. Every minute helping a newbie is a minute spent away from writing code. The high turn-over rate of contributors combined with programmers who say that they want to help and then disappear, makes senior developers cautious about choosing who to help.

One of the first things that all interviewees stressed as being important in assessing the potential of new contributors was the chosen entry point, and their initial introduction on the mailing lists. This is what Sebastian, a senior KDE developer, had to say on this issue:

> Yeah, you realize that from the first step he takes. If he is like coming to mailing lists saying, hey, how can I help, that's the kind of people that are not going to follow up and I am actually on the how to help alias for KDE org, so I get all those mails from people that say, hey, how could I help and one percent remains up there, even though it try to make very helpful answers and guide them to what they could do, they don't go anywhere, because those people assume that there is an easy answer to a question of how can I help and that someone is going to teach them everything and that's not how it works, because no one is going to take the time to do that. On the other hand if you see someone coming to a mailing list with already a patch to fix something or you feel from his questions that he has really looked into things deeply, then you know that it is going to be someone who is going to be efficient and that you can spend a bit of time helping that person.
>
> (Sebastian, 23/08/04 p. 14)

Sam, another senior KDE developer, describes the rules of thumb experienced developers employ in assessing the level of commitment of new developers:

> I think that most of the time, its not as immediate, so, you can't talk to them once and tell whether they are going to stay around, but if they, its very obvious if someone shows up and then in the first two weeks you start see productive results out of them, you can say, ok, its worth to invest more time in this person, they are looking like they are regularly contributing things back or having some sort of sustained productive output, whereas there are some people, some people very much like the idea of being an open source developer, but they are not willing to put the work into it and its a lot of work.
>
> (Sam, 14/04/04, p. 15)

Martin, a GNOME developer, went further, pointing out that ideally new participants should pass through the stage of peripheral participation very quickly, discreetly and almost invisibly:

> No, new developers that come along, the good developers suddenly appear on the scene as if they had been working on the stuff for years. They will have figured all this stuff off very, very quickly and I suppose it is this kind of developers that we really want to attract. So we are not trying to really attract, people who are not capable of going off and figuring out stuff for themselves. So it is almost like a baptism of fire, you just think or you swim, so it might be a little bit cruel, but the amount of effort that would be required for an existing developer to bring a new developer up to speed who isn't capable of bringing himself up to speed, you might not get a good developer out in the end, at the other end, which is, you know what I mean?
>
> (Martin, 11/07/04, p.4)

Martin's comment indicates how intertwined the values of self-reliance, commitment and productivity are. Putting in the time and effort to find out things for oneself is an indication of commitment and, at the same time, a prerequisite for sustained participation; successful information seekers and dedicated learners do not impose on the time and attention of senior developers and the incremental self-reliant development of their knowledge, a common characteristic of experienced developers, attests to their potential as productive contributors. Although the importance of learning in F/OS is acknowledged, peripheral participation is meant to be a solitary activity, taking place in the background and not at the forefront of developments. One interviewee mentioned that it was important in designing peripheral activities that involve coordination with developers to ensure that they fitted into existing routines. This interviewee gave an example of the success of KDE junior jobs, of bugs earmarked by maintainers as suitable to be tackled by new contributors as they are submitted to the bugs database. In addition, as Sebastian indicated, seasoned developers usually judge the potential of new contributors very quickly, sometimes even from their first couple of postings. How potential contributors introduce themselves to the community is not only related to successful face-work, a sign of whether or not they have successfully assimilated the behavioural 'scripts' of F/OS development but also a newbie's initial postings usually indicate the extent to which they have already committed to the development process. In addition, as Sebastian indicated, seasoned developers usually judge the potential of new contributors very quickly, sometimes even from their first couple of postings.

Nevertheless, the imperative of production can have a negative impact on participation. In his keynote speech in GUADEC 2004 (GNOME Users and Developers Conference Nat Friedman commented on the barriers to participation and innovation in F/OS. He suggested that the F/OS culture has become intolerant of newbies and that barriers to entry have become too high, hindering the influx of new ideas and compromising the future of the community. As indicated in Chapter 4 the commercialization of F/OS projects may be one factor responsible for this mindset. Balancing the need to respond to the demands of programs being utilized in production environments, and the requirement to attract and incorporate new volunteers appears to an ongoing issue in the context of F/OS projects. Section 5.4 highlights another aspect of peripheral involvement which concerns the participation of non-coders. If the participation of coders is underlined by the imperative of production the contributions of non-coders seem to be guided by the desire to facilitate and promote access.

5.4 Autonomous peripheral participation and development

This section focuses on autonomous peripheral participation, a term used to describe the various activities that support and complement code development within F/OS projects. Within the context of GNOME and KDE the main supporting activities are: translations of the available documentation and, most importantly, of all program and interface aspects that are visible to the user (such as application and contextual menus, error messages, dialogues, help information, etc.); writing documentation for users and developers; usability related work (improving user interfaces); accessibility (providing alternative means of interface for handicapped users); and marketing and artwork (creating logos, screensavers, wallpapers, splash screens, icons). The analysis described in this section is based primarily on an investigation of the translation and documentation teams.

Section 5.4.1 focuses on the distinctive characteristics of peripheral teams such as their organization, the way that they position themselves within development and their view of projects. Section 5.4.2 investigates issues relating to cooperation between coders and non-coders, and highlights some of the arrangements and processes in place to facilitate their collaboration.

5.4.1 Non-coding teams: organization and participation among translators and documenters

Translation and documentation teams within GNOME and KDE are similarly structured. In both communities there are over 50 different internationalization (translation) teams, generating translations from English, to German, to Welsh, to Swahili, and involving more contributors than coders. Documentation teams are much smaller and usually involve a handful of regular volunteers.

Both KDE and GNOME have made significant efforts to lower the barriers to participation, especially in terms of the technical skills required, and to automate the process to a degree that, in theory, even people with no knowledge of Linux and Unix can contribute to the internationalization effort. Both projects, for example, have developed, specialized translation tools, such as Kbabel, an application originally developed for KDE by a coder interested in translations. Among other things, this application allows translators to create a database with all available translations that helps them run searches to identify which strings have been translated and obtain recommendations about how to translate specific terms. GNOME translators can draw on a collection of scripts known as Intltool, which simplify the process of using the specialized editor Get Text employed in the translation process. In addition, a number of tools and platforms have been developed to facilitate and coordinate translations between and across different groups and applications via a Web based interface. One of the most prominent examples of this is Rosetta (https://launchpad.net/rosetta), a Web based translation system that was initially developed for the Ubuntu distributions, but which also supports the activities of other translating groups.

In certain cases, however, the low level of entry and expertise associated with peripheral development appears to create problems in ensuring the quality of contributions. This is what Ken had to say on the subject:

> Yeah, a lot of people don't want to make any kind of commitment. They want to do something one time, get their buzz that the helped out on a open source

> project, I am cool and that's it, they don't ever want to see it again and that's fine with me (laughs). If the content is good. It seems to me that those kind of people are sometimes more trouble than they worth because they are also the ones that tend to dust something off in five minutes without putting a lot of thought into it and it is quite sloppy and needs editing and massaging to be put into the document and then when you say that to them or you send a draft back for them to check over they are like, God, I gave you this thing for free, just put it in the user manual. Your user manuals are terrible, you need my help! Yeah, they are doing us a big favour, but well, they are really not, if their five minute contribution takes an hour of work from my part to shape into something we can actually use, I may as well have written that myself.
>
> (Ken, 01/10/04, pp. 10-11)

The internal organization of the translation teams resembles in its variety that of coding teams, but there are some important differences. In both KDE and GNOME the language team is required to have a leader whose role is similar to that of a maintainer. Language team leaders are responsible for correcting mistakes, reviewing, and integrating the different parts of the translations by ensuring that the agreed terminology is used consistently. The internationalization projects have an additional coordinator responsible for organizing the new language teams and liaising between coders and translators. Also, whereas modules, in some cases, can have more than one maintainer, each language team has only one team leader and usually only one or two people with CVS commit privileges.

It is generally not considered necessary for all translators to have CVS accounts, although people's views about this differed. Two people interviewed for this part of the study suggested that it was important for all regular contributors to have CVS commit privileges in order to ensure accountability. However, the more dominant view was that translators did not need to have commit privileges. There are several reasons for this. One interviewee who was the coordinator for GNOME's Internationalization project suggested that the reluctance to assign CVS commit privileges is associated with the higher turnover of translators than coders, which is perhaps due to the lower entry requirements. It could also be related to the fact that creating CVS accounts is no trivial administrative task or because translators as a group have a lower level of technical expertise than coders and are considered to be more likely to 'mess up' the development tree by leaving the submitted files in an unused state. Limiting the number of CVS accounts is a defence mechanism in this case.

One of the most important requirements for language maintainers in both projects was to subscribe to the internationalization mailing list. Although most language teams have their own mailing lists, the internationalization mailing list is the translators' main forum. It coordinates the two communities by informing translators about new policies, release schedules and important project developments. It is also the form for translators to discuss terminology and general project and translation issues. One of primary tasks of the team coordinator is to keep abreast of the internationalization mailing list and filter relevant information back to the team.

Peripheral contributors described themselves as users and insisted that their work had a more practical orientation, and facilitated access and dissemination rather than directly promoting development. In fact eight long-term interviewed peripheral contributors[47] did not appear to be interested in coding. This is what Ken, a leading documenter and translator in the KDE project had to say on the subject:

> It is because we are users, much more than the developers, we are the ones that open every single application and look at every dialogue, we are the ones that actually have to use them, and it is also the main impetus for the translators to get involved, it is because they want to use the software in their own language. Most of translators, that's the main reason that they are doing it. We tend to be a lot less of the hardcore free software kind of people. Most of us are doing it for really practical reasons, that a simple way to put it and I am generalizing. But, yeah I think it is mostly true.
>
> (Ken, 24/08/04, p. 3)

The requirement to use the applications in order to translate them correctly has an interesting implication. It has been pointed out that whereas many maintainers concentrate on their own modules and areas of development, translators through their work can obtain a global view of the project. Sean, a translator in GNOME who previously was involved in bug tracking, described it as follows:

> I think contributing on the periphery, not in the code, means that you end up with a better overview and understanding of the project as a whole and, for example, if you are going to Bugzilla, even if you think that you are only looking for bugs in the file manager. One bug that you deal with may be related to underlying libraries and how they deal with information and so you learn about that and then the next bug may be related to the look of the thing and you will have to go and look at the human interface guidelines, which are the usability aspect and so you learn about that and similarly you go through proof reading documentation or translating strings and you look at a string perhaps and say "oh, wow, I have got to translate that, what program is this?" And you will often look and you realise that you had no idea that that program could do that and you learn this about all the programs, not just this that you are coding on. I have seen discussions on the mailing lists, about I think that it would be great to add this kind of functionality to this application, says somebody who works on this application and then somebody else would say. Why? We already have it over here on this one. So I think that at the edge, you get a much rounder, overall picture of all the little bits going on.
>
> (Sean, 09/03/04, pp. 19-20)

Despite the fact that translators are ideal beta testers, the possibility that peripheral knowledge will feed directly into development does not generally materialize. There appear to be three main reasons for this. Firstly, lack of interest on the part of many translators in participating actively in development might explain why they are do not routinely report bugs even for stable releases. Secondly, the degree of technical knowledge required to compile programs from their sources, which is, as already indicated, a prerequisite for participating in real-time development, appears to be a stumbling block for the technically less sophisticated

[47] This includes two peripheral contributors interviewed for KDE Quality teams.

peripheral contributors. Thirdly, (and we explore this further in the next section) the translators' patterns of work and collaboration are intensive and seasonal.

The case of documentation is slightly different; more sophisticated skills are required for technical writing. Documenters in F/OS source have the difficult task of producing documentation suited to end-users and prospective developers. In addition, in documenting different project modules writers often cooperate closely with developers, who may have documented some aspects of their projects, but often not to the standard required of professional technical writers. However, although significant effort is being made to make bug reports and mailing lists accessible through search engines, this information is not usually integrated in the project documentation. This seems mostly to be due to the small number of regular volunteer documenters working in GNOME and KDE.

Thus, the tools that the internationalization teams are provided with reflect the desire to lower the barriers to participation for translators. In certain instances, however, the lower level of entry and expertise associated with autonomous peripheral development creates problems with regard to ensuring the quality of the contributions. The organization of translator teams is underpinned by certain minimal requirements, such as the need to have a team leader who keeps track of development through the main internationalization list. Although there are possibilities for closer collaboration between coders and translators, the opportunities for peripheral knowledge to feed into the development are only marginally explored. In addition, the positions adopted by translators and coders within the community differ. The discourse of participation of translators is underlined by notions of access and dissemination rather than technical excellence and competence. The case of documenters is different, since contributing to this aspect of development requires a higher level of specialized knowledge than is required for translation.

5.4.2 Seasonal workers, distant relations: cooperation between coders and non-coders

This section looks at coders' perceptions regarding the role of translators and highlights the requirements and the processes set in place in order to facilitate collaboration between these teams.

Despite the existence of the main internationalization mailing list and the requirement for team leaders to follow the postings made on it, coders often complain that it is difficult to get messages through to translators. Ken, a KDE contributor who is primarily involved in documentation, but who has also had considerable involvement in translations, commented on the problems regarding communication between the two communities and stressed the importance of the coordinator, who needed to be someone that could straddle the boundaries of the two groups:

> I think that is actually a problem; we used to have Thomas Diehl, being the guy between and he has definitely stepped back and nobody has taken up his place,

> to actually, this is what I do for the docs, it is my job, its just getting in there and talking to people to relay messages back and the translation team could really benefit from somebody doing that, because the developers just don't think, don't remember to ask the translators and the translators seem to sit in their little ghetto on the mailing lists and complain about things to each other but don't actually go to the developer and every time it happens, some random person will tell the developer, there was a really big problem with the application and they come on the mailing lists and say, oh my god I am so sorry what can I do to fix it and it is immediately fixable and I think having a coordinator would really help but it is a big job and it really needs someone who is really willing to put in a lot of time to step up and do it.
>
> (Ken, 1/10/04, p.1)

The particular problem that Ken refers to occurs when programmers fail to mark up the strings of the code that are visible to the user and therefore need to be translated. Translators use specialized tools, such as Intltool's Get text, in order to extract these translatable strings and compile the special (.po) files that are used for translations. In finalizing their work, translators need to integrate the translated strings back into the development tree. If the marking-up of the strings to be translated is not done, or is not done adequately, then the translators cannot do their work. This is one of the basic requirements that coders must take into account when writing their applications.

Another practice that has been developed to accommodate the needs of the internationalization project is that of 'string freezing'. This is the last in a series of accumulative freezes[48] and generally involves a period of some two to three weeks prior to each new release during which time hackers are not allowed to make any changes that might affect the string of messages in the code that are visible to the users. This allows translators to catch up, complete and update the translations of messages that up to that time have been continuously changing as applications are worked on. Similarly, this period gives documenters the opportunity to finalize the documenting of new features and any other improvements made to the code base. Thus, unlike coders who can work more or less constantly on the project, translators are 'seasonal workers' whose involvement in the development process is for a couple of weeks before a major release during which time the bulk of their work is accomplished.

The imposition of the string freeze on coders is not unproblematic. Knowing that there is considerable time before the release when enhancements could in theory be made to programs, is, as suggested in Chapter 4, section 4.4.2, frustrating for many coders. However, communities have made provision for the string-freeze to be broken

[48] The first freeze is usually referred to as the 'feature freeze' and involves finalization of the modules and new features to be included in the release. The second freeze concerns the libraries, the deep level code that underlies all modules. The third freeze usually concerns the user interface (UI). The string freeze is the last freeze before the 'hard code freeze', i.e. the final freeze period for translators and documenters to work and the period when minor bugs can be fixed. For more information on the various release phases and freezes of GNOME and KDE, visit: http://live.gnome.org/ReleasePlanning/Freezes and http://developer.kde.org/development-versions/release.html, last accessed 04/03/07)

and for developers to make alterations to their programs within that period. For this to occur developers must a make case to the translators' coordinator and the release team (GNOME) or the release dude (KDE) as to why the proposed changes are necessary. Decisions are generally made on a case-by-case basis, but it is not terribly clear who actually authorizes the string-freeze breakage.

Most translators and coders involved in the internationalization project, seem to think that, despite the progress that has been made in accommodating the needs of translators, programmers need to be constantly reminded of their presence and the requirements for their collaboration. This is largely due to the relatively low importance ascribed to peripheral activities compared to contributing code. Although the contribution of non-coders is increasingly appreciated, 11 interviewees agreed that programming is undoubtedly the most is valuable form of contribution and that developers usually have more say in the development process. Al, a developer in the GNOME project, explained this further:

> To a certain extent in a lot of open source projects and in GNOME, it is definitely true that the people who contribute the code are the people who get to decide what happens. Now this kind of may lead to some technical citizenry in that the people who do the graphics or the people who do the documentation do have to work quite a lot harder to be seen as a really key contributor who can help make decisions and I don't think that there are many people would include in listings as core developers would be someone who wrote documentation or did artwork or something like that, which may, well I guess you might say, well they are not developers but they certainly have a role to play there.
>
> (Al, 24/06/04, p. 13)

To sum up, cooperation between coders and translators is highly modular, and communication between them appears to be fragmented. This seems to be the result of a lack of interest on the part of the two groups about the other's work and a consequence of the way the internationalization effort is organized. Documenters and translators are seasonal workers who are involved in the development process only at certain periods when the work of the coders is on hold. Although there are no easy answers to how the process could be better organized in order more effectively use the time and effort of peripheral developers and the knowledge produced at the periphery, this is likely to become more pressing as F/OS projects improve their links with the non-technical user base. This is explored in the next section.

5.5 *Autonomous peripherality in context*

While attracting new developers is critical for all stages of F/OS community growth, how to encourage and better organize autonomous peripheral participation seems to be more closely associated with the evolution of F/OS ties with the wider user and developer base. As F/OS becomes increasingly incorporated in non-technical production environments, that is, through exploitation on home and office desktops, the need to understand and support non-technical users also increases. Although this is considered to be primarily the purview of F/OS companies, as one of the main aspects of their business model, communities are showing an increasing interest in involving non-coders in the development process (for example, see quote introducing the idea of KDE Quality teams in section 5.5.1). In addition to addressing more directly the requirements of the non-technical user base, this interest is also related to the desire to signal a strong sense of community and the maturity and user-friendliness of the project.

The large-scale deployment of F/OS in companies, NGOs (Non Governmental Organizations) and public administrations appears to make these issues more pressing. This section highlights some of the solutions that communities have experimented with for supporting and organizing autonomous peripheral activities. The main case study that is drawn on is that of the KDE Quality teams (see Chapter 3, section 3.5.1.3), an initiative aimed originally at creating a supportive non-coding community around the various applications of the KDE platform.

5.5.1 Examples of organized peripheral participation within and across community boundaries: KDE Quality teams and the Open Usability project

This section provides an overview of the KDE Quality teams, an initiative that aimed to organize autonomous peripheral participation in KDE. The initial aims of the project are set out and the challenges and subsequent changes made to its focus and orientation are described. Comparisons with the Open Usability project enables some conclusions to be drawn about the challenges involved in organizing autonomous peripheral participation.

The idea behind KDE Quality teams (http://quality.kde.org/, last accessed 27/03/06) was introduced in 2003 by Leonhard Carlos Woeltz, a trade investment consultant who became involved in F/OS because of his belief in its potential for developing regions and countries. This is how Woeltz initially conceived the role of KDE Quality teams, which initially were referred to as Janitors:

> The main idea is to create a community based quality team of non developers, that would focus on the whole of individual modules of applications, working orthogonally to developers, documenters, users and testers, instead of the specific of the whole. In other words think of acting upon the whole of Kontact instead of acting upon the what's this of KDE project. The key idea is attracting people in a way that's both interesting to them and more useful to KDE project. This would be the basis of a community oriented (instead of company oriented) effort of improving this experience. We have a wonderful community, kde-look.org, KDE wiki and all the translating teams are strong evidence of this.
>
> (http://lists.kde.org/?l=kde-usability&m=107187409020725&w=2 last accessed 27.03/06)

The aim of the KDE Quality teams was to better organize the increasing non-programmer involvement in KDE by creating a supporting group around each application, in the form of a team of contributors providing help to maintainers in the areas of design, user interface and documentation. The local character of the teams and their attention on a single project were expected to provide a clearer focus and increase the 'social experience' of participation (see http://lists.kde.org/?l=kde-usability&m=107187409020725&w=2 last accessed 27.03/06). At the same time, the desire for a community rather than a corporate based initiative is indicative of the wish to maintain the grassroots character of peripheral activities. Such activities were seen as having the potential to enhance the appeal of non-coding group involvement, often seen as an indication of the strong sense of community of such projects. This is seen as valuable for attracting new volunteers, both coders and non-coders.

Although the community welcomed the initiative, the project faced a number of challenges which changed its original orientation. One of the main problems faced by its founding members was the difficulty involved in communicating the aims of the initiative in a clear manner - many potential contributors thought that in order to be able to join these teams they needed to be experts in documentation, artwork and usability. A second problem was the time and effort required to set up teams - it seems that many potential volunteers were discouraged by the sheer amount of work that was involved. Eventually only a handful of Quality Teams was created, the most notable being the KDE PIM (Personal Information Management) Quality team, whose success depended largely on the presence and continuous effort of Leonhard Carlos Woeltz. A third challenge related to the difficulty in eliciting the help of coders for certain tasks, such as creating and maintaining task lists.

Faced with these problems the project focus shifted and activities related to supporting new contributors were prioritized. KDE Quality Teams gradually became a conduit, the first point of contact to which new contributors could be steered to familiarize themselves with the structure of the project. Help for newbies - coders and non-coders - came in the form of documentation and guidance provided through the KDE Quality teams' mailing list. GNOME has a similar mailing list - 'GNOME-love' - dedicated to newbie queries. Groups of activities for which the Quality Teams were supposed to act as an umbrella, as in the KDE Usability Project, developed into separate projects. In the case of usability, the situation changed dramatically following the involvement of Relevantive AG) (http://www.relevantive.de/) a German based usability company that offered its services to the KDE and F/OS software project pro bono. The company created a separate project called the Open Usability Project (http://openusability.org/projects/kde-hig/), which provided a host of services, including direct expert advice for maintainers that wanted to improve the user interface of their applications. The involvement of a company specializing in usability seemed to alleviate a host of issues for maintainers. In an interview in 2004 predating Relevantive AG's involvement in the project, Dean, a KDE coder described the challenges of usability and of involving non-coders in the development process:

> A lot of the time when we are designing interfaces, in the past we were able to design them from the point of view of how the technical person drives this and what is changing now is we need to much more take into account how a non technical person drives it, but we need to do that without upsetting the technical people.

> Interviewer: Well, that's tricky (laughs)
>
> Dean: Yes, it is very hard, the problem is that the technical people are likely to be the people who would be the next generation of developers who take on board. To say that they don't like the project, we don't get any new developers and ultimately the project will fail. We have a difficult balance to strike and at the moment, you know, that's one of the big questions in KDE and if you look at KDE developers.org, the sort of the web blog site, you find that there is a lot of discussion going on about how we can manage this process and integrate teams of people who are trying to improve the user interface with the teams of people who are trying to develop the system without everybody shouting at each other and eventually storming off.
>
> (Dean, 22/03/04 pp. 8-9)

Dean's comments highlight two of the issues most central to the organization of autonomous peripheral participation in F/OS. The first relates to the question of how to involve the non-technical community without alienating the 'techies'. The second relates to how to improve the cooperation between them. The KDE Quality teams and the Open Usability project provided two very different solutions to these problems. The more local character of the KDE Quality Teams with their focus on individual applications rather than on the whole of the project, was based on a holistic approach, aiming to encompass a wide range of activities within the context of the same sub-project. Furthermore, the initiative was addressed to everyone who wanted to become involved, regardless of their experience or expertise. The Open Usability project had a narrower focus, but the professional credentials of the actors behind it helped diffuse subjective arguments on what improves user experience, and the provision of consulting services on a case-by-case basis was less intrusive for maintainers. Lastly, whereas the KDE Quality teams was positioned as a KDE specific initiative, the Open Usability project provided its platform and services to all interested F/OS developers.

5.6 Conclusion

This chapter has investigated two different types of peripherality. The first, the legitimate peripheral participation of new developers, is consistent in its focus with the CoP perspective since it involves patterns of joining and the participation of new community members as they improve their skills and move from the project's periphery to its centre. The second type of peripheral participation, autonomous peripherality, focuses on the participation of non-coding teams and highlights their distinctive characteristics and patterns of contribution as shaped by their specific agendas and their interactions with the main development community. This section draws together the insights that emerged from the analysis and discusses their implications for understanding the dynamics of development and cooperation in community led F/OS projects.

Despite the fact that the architecture of F/OS projects, such as their high degree of modularity, is supposed to facilitate learning and participation, the views of new developers highlighted various barriers to entry to F/OS. These included the requirement to become familiarized with the development tools, the need to form a conceptual understanding of the program's architecture and the rules involved in committing code, and the difficulties associated with situating oneself in the development.

Although the importance of helping out new developers is generally recognized, and attempts to organize and facilitate their integration are reflected in the existence of mailing lists specifically set up for this purpose and the provision of tutorials and documentation, it is clear that peripheral participation takes place in the background, not at the forefront of development. New developers are generally expected to orient themselves by making do with whatever learning resources are available and gaining a working understanding of the project before seeking the help of experienced developers.

Investigation of the dynamics of cooperation between senior and new developers suggests that the role of learning in F/OS communities goes beyond that of establishing a common framework of shared values, practices and networks of contacts between peripheral and central members. Analysis of the interviews indicates that learning processes are integral to the exercise of power and control. The significant barriers to entry are seen by senior developers as elements in a process necessary to ensure a certain level of commitment and capabilities in new contributors.

The analysis suggests that there are inherent tensions between the need to attract and integrate capable volunteers and the demands of continuous production. F/OS are not simply communities set up as locales of knowledge management; they are primarily communities organized around the production of a complex good - software. In many ways the views of senior developers on peripheral participation echo Goody's (1989) remarks regarding learning within the context of an apprenticeship. According to Goody apprenticeship presupposes involvement in real productive work. As a result learning within its context is scaffolded by the technical constraints of production, which, in turn, demands that the apprentice becomes integrated in the production process as safely, but also as quickly as possible. This aspect of CoP has been somewhat neglected in the literature.

A significant differentiating factor in F/OS compared to traditional apprenticeship contexts concerns the minimum degree of commitment demanded for a meaningful contribution, which is often undermined by the high degree of turn-over of contributors, prompted by the appeal of being known as an F/OS developer and the ease of signing-up in an online, volunteer community. As a consequence, it seems that the criteria for being recognized as a potential valuable contributor, a legitimate peripheral participant, in F/OS differ substantially from those in offline professional networks and CoP where institutional frameworks, such as formal employment relations, formal accreditation schemes and tighter social networks, ensure a certain skill level and some degree of continuity and commitment.

Some of these criteria in F/OS include assessment of the new contributors' potential based on their chosen point of entry and the way that they position themselves within the community. New developers who seek help on specific issues having demonstrated an active engagement with the project are generally considered more promising than newbies asking for general help and guidance. The speed with which senior contributors appear to judge potential contributors is consistent with Stewart's (2004) remarks on the swiftness with which F/OS communities decide on a member's position in the project's hierarchy.

Analysis of the characteristics of the translator and documenter communities, and their relations with coders, highlights another aspect of peripheral participation. In this case the findings indicate that contributors in areas that support and complement code development position themselves in the community differently from coders, prioritizing access rather than technical excellence. The organization of non-coders' participation in the development process is structured by an effort to lower the barriers to participation and minimize disruptions to the work of coders. As a result, the work-flows of the two communities are organized in a horizontal rather than a vertical fashion; the two groups operate in relative independence with some key members acting as go-betweens. The bulk of the work of translators and documenters is carried out during the string-freeze. The case of documenters varies from that of translators in that their work requires closer interaction with developers and a degree of technical knowledge.

These organizing factors combined with a relative lack of interest in each other's activities and intensive development mean that peripheral developers' insights, such as their global view of the project and their experience as power users, remain largely unexploited. This creates some interesting questions not only with regard to the participation of non-coders, but also with respect to the participation of non-technical users and the way their feedback is utilized. This chapter has shown that the characterization of F/OS communities as horizontal networks of innovation by and for users (von Hippel, 2002) needs some rethinking to take account both of differences among different types of users and relations between the various groups of contributors participating in F/OS projects.

The investigation of autonomous peripherality indicates that rather than forming a homogenous community of development, collaborative communities focused on the production of complex goods constitute constellation of practices, joined by the project's overarching goals, but distinctive in their priorities, rhythms of development and patterns of collaboration. Although the types of contributors involved may vary depending on the targeted application user groups, the growth of F/OS projects and their adoption by wider user and developer audiences suggest that autonomous peripherality may become increasingly important for many communities.

It appears that in some cases lessons have been learned. F/OS projects that have been initiated by corporate actors allocate explicit resources to document writing, and certain initiatives, such as the Rosetta platform[49] move to a greater degree of integration of translations across and between different open source projects.

In the course of the analysis described in this chapter it became clear that although activities, such as writing documentation, are directly associated with the recruitment and learning process, other aspects of such autonomous peripheral participation are also important, not only with regard to their immediate benefits for the project's development, but also because they are often seen as a sign of a project's maturity and success. A vibrant usability, artistic and translation community may also attract volunteers. However, this is not the only point where the two types of peripherality converge. Both forms of peripherality appear to be integral aspects of the division of labour in F/OS communities; whereas one regulates the division of labour within the context of the same team, the other expresses the dynamics of labour division among the various groups of contributors to the projects and is indicative of the value that is attributed to different skills and knowledge within the community.

The analysis indicates that these two types of activity are regulated through two different modes of organizing. The first relies on embedding modes and flows of cooperation and interaction within the context of everyday development, for example, the string-freeze period or the mark-up of easy-to-solve bugs by experienced developers. The second is associated with the development of distinct initiatives that disembed these activities from their original development context. This applies to the KDE Quality teams and the Open Usability project. Unlike the KDE Quality teams, which focused on the KDE, but similar to the Rosetta platform, the Open Usability project aimed to address the needs of not one, but several, F/OS projects. Initiatives like these are indicative of the growing importance of peripherality and the need to streamline and pool efforts and resources from different areas of expertise.

These different modes of ordering and organizing labour within F/OS projects can be regarded as constituting different 'technologies of communities' mobilized within their context. It could also be argued that F/OS projects and other related initiatives are themselves assemblages of these arrays of different techniques, such as the established systems of differentiations that form part of these technologies, for ordering relations of cooperation and production. The notion of 'technologies of communities' originates from Nikolas Rose (1999) who used it to describe the various techniques, such as market research and opinion polls, that have been devised to make communities real (see Chapter 4, section 4.6).

[49] Rosetta is an Internet platform for open source application translation. It enables people to translate their favorite open source application into their spoken language (https://launchpad.net/rosetta/).

It is proposed here that whereas the idea of a meshwork is useful for disentangling the web of relational power, the notion of 'technologies of communities' is useful for understanding how different practices and discourses are ordered through the use of specific tools and techniques that not only help visualize, order, map and survey community space, but also organize relations of production. Figure 5-1 depicts the main themes and concepts explored in the sections of this chapter and the connections between them. The underlying dynamic of Chapter 4 concerned the balance between the gift and the exchange economies, whereas in this chapter, the overarching theme is related to the division of labour in F/OS. Chapter 6 begins to draw these two aspects together by indicating patterns of contribution at the level of non-affiliated and affiliated maintainers and members of the GNOME Foundation and KDE e.V.

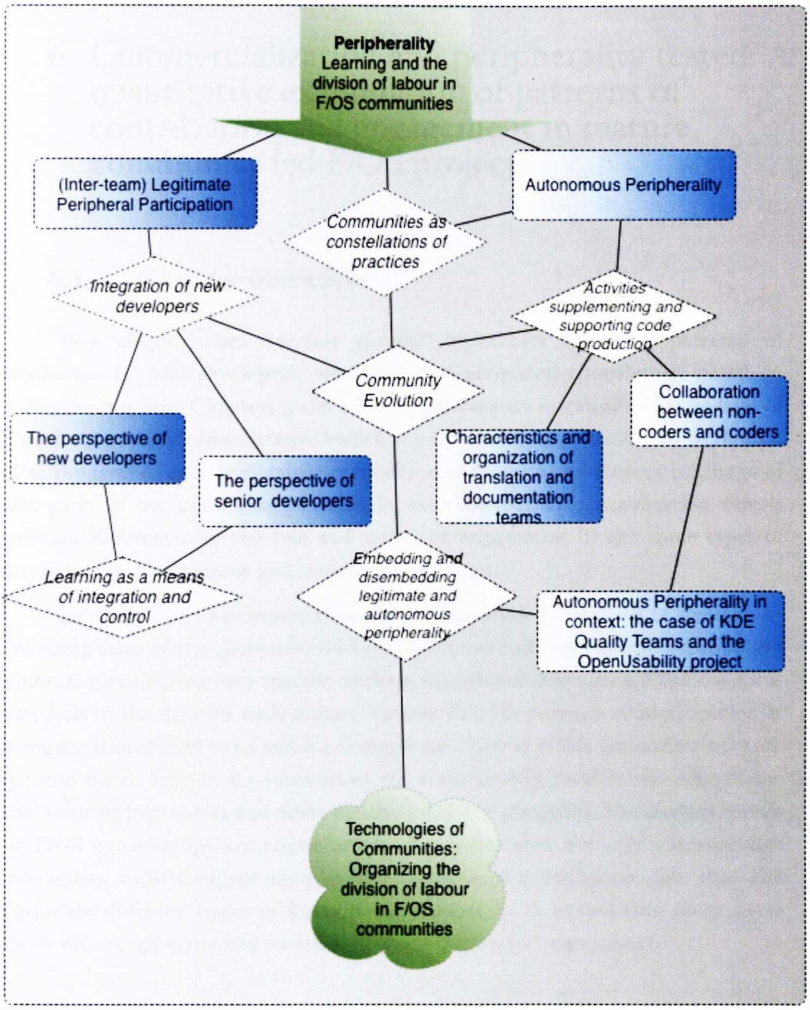

Peripherality
Learning and the division of labour in F/OS communities
(Inter-team) Legitimate Peripheral Participation
Autonomous Peripherality
Communities as constellations of practices
Integration of new developers
Activities supplementing and supporting code production
Community Evolution
Collaboration between non-coders and coders
The perspective of new developers
The perspective of senior developers
Characteristics and organization of translation and documentation teams
Embedding and disembedding legitimate and autonomous peripherality
Learning as a means of integration and control
Autonomous Peripherality in context: the case of KDE Quality Teams and the OpenUsability project
Technologies of Communities: Organizing the division of labour in F/OS communities

Figure 5-1: Peripherality key research themes and their interconnections

6 Commercialization and peripherality tested: A quantitative examination of patterns of contribution and engagement in mature, community led F/OS projects

6.1 *Chapter overview*

This chapter aims to test specific hypotheses regarding patterns of contribution by paid developers, volunteers and peripheral contributors based on insights derived from Chapters 4 and 5. These patterns are investigated at the level of the projects' primary administrative bodies, the GNOME Foundation, and KDE e.V., and at the level of their maintainer networks, the groups of developers in charge of those parts of the code base included in each release. This examination reveals significant differences in the role and modes of engagement of the three types of contributors in development and community processes.

Section 6.2 of this chapter elaborates a framework for analysis of the data by formalizing some of the qualitative findings of the research on commercialization and peripheral participation into specific working hypotheses. Sections 6.3 and 6.4 focus on analysis of the data for each project, looking first for patterns of participation at the organizational level (6.3.1 and 6.4.1) and in the context of the maintainer network (6.3.2 and 6.4.2). Section 6.5 summarizes the main findings, and in the light of the initial working hypotheses and draws out their wider implications. The analysis reveals that F/OS communities are constellations of practice that not only accommodate communities with divergent agendas and patterns of contribution, but that also incorporate different layers of institutional support. It is argued that these layers provide diverse opportunities for learning, participation and engagement.

6.2 *Working hypotheses for the quantitative examination of patterns of contribution and involvement*

As indicated in Chapter 3, the qualitative aspect of the study, namely the interviews and the ethnographic observations, is closely intertwined with the quantitative methodology used to conduct an examination of the two communities. Chapter 3 introduced the conceptual framework for this approach by outlining the broad questions being addressed in this investigation, including the ratios of volunteer and paid contributors, coders and non-coders in the two communities' principal organizational bodies. This section formalizes the main qualitative findings on commercialization and peripheral participation as a basis for the analysis of the quantitative data.

Chapter 4 indicated that paid developers are frequently identified with the core developer group. This group appears to be characterized: by the large number of commits across the spectrum of development, including the infrastructural elements of the project, and by a strong presence in other aspects of community life, expressed, for example, by participation in conferences and various administrative and technical bodies, etc. as representatives of the community In particular, ten interviewees suggested that that paid programmers, especially those with close community ties, are likely to contribute to, as well as to maintain many critical parts of the code base, such as those associated with programming libraries and other essential elements of the projects platforms and desktop components. As suggested in Chapter 4, this appears to be closely linked to corporate recruiting strategies. Companies often hire developers who they estimate are on a 'critical path', that is, individuals who have already demonstrated a high degree of technical competence and who already have involvement in the most critical and, often difficult, aspects of development. This study is not able to shed light on the precise direction of this relationship nor, in particular, on whether a person was hired because he or she was already a core contributor or becomes a core contributor through his or her work. However, the data do provide insights into common and divergent patterns in the contributions of paid and volunteer developers.

A second point of convergence in the interview findings was in relation to the distinctive nature of the communities of developers and peripheral contributors. The chapter on peripheral participation (Chapter 5) indicated that, although the two communities share a common underlying agenda in terms of their interest in F/OS, the groups have different priorities: whereas the developers focus on technical excellence, peripheral contributors are more interested in access and practical use. This chapter shows how these priorities are associated with the patterns of contribution of the groups, and examines whether or not programmers become involved in documentation and translation and whether peripheral contributors move beyond their initial group boundaries.

Similar to the case of volunteers and paid developers, an additional factor that can provide an indication of the cohesiveness, or lack thereof, between peripheral contributors and coders is participation in subgroups and community events. The definition of meritocracy on the basis of code commits (see Chapter 5) and the two different priorities of the two groups suggests that peripheral contributors, although not precluded, are less likely to participate in high-level community steering bodies. Although this study does not investigate the extent of involvement of non-affiliated and affiliated contributors in the leading technical and administrative committees of projects, it does examine whether there are different rates of participation in conferences by analyzing the participation of peripheral contributors and coders, volunteers and employees in GUADEC and the KDE akademy, the major community events of the two projects.

To summarize, analysis of the qualitative data allows us to formulate four working hypotheses:

1. Paid[50] developers are more likely to contribute to critical parts of the code base.
2. Paid developers are more likely to maintain critical parts of the code base.
3. Volunteer contributors are more likely to participate in aspects of the project that are geared towards the end-user.
4. Programmers and peripheral contributors are not likely to participate equally in major community events.

6.3 The GNOME community

This section presents the findings for the GNOME project which confirm the four hypotheses about the central role of paid developers and the distinctiveness of the volunteer and peripheral communities. Moreover, analysis of the data reveals interesting differences between paid contributors employed to work on GNOME and those employed to contribute to other F/OS projects. Section 6.3.1 provides an analysis of the GNOME Foundation network; the analysis in section 6.3.2 on the GNOME maintainers, is less detailed because the data on maintainers did not reveal additional insights, but rather confirmed the view that emerged from analysis of the GNOME Foundation members' survey responses.

6.3.1 The GNOME Foundation network

As indicated in Chapter 3, the survey gathered data from 59.4% of all members listed in the GNOME Foundation (335 in June 2005). We first present some basic data concerning the split between volunteers and paid contributors and a brief outline of the organizations employing them. This is followed by an overview of the different groups of contributors in the Foundation, based on a ***hierarchical cluster analysis*** of the respondents and the ***areas of contribution they reported being active in during their involvement in the project***. These two groups of findings are compared with patterns of contribution among volunteers and paid developers based on their ***main area of contribution***, which, in this study, was defined as the module/project that they reported contributing to ***most intensively during the six months preceding the*** survey (see Chapter 3 for the methodology). Next, the findings related to *GUADEC attendance* are presented. Finally, several of these variables are combined in a *logistic regression model,* which fine tunes the examination of previously detected significant associations.

[50] 'Paid' and 'affiliated' refer in this context to developers employed by an organization actively involved in F/OS development. A volunteer is a contributor who is not employed by such an organization. This does not mean that all volunteers are unemployed, but simply that they are not remunerated for their participation in F/OS development.

In this study the respondents' reported module/project contributions are grouped into five distinct hierarchical areas of development. The coding scheme was described in detail in Chapter 3, and its basic principles are outlined below. The five areas of development are:

Core/Platform modules. These comprise the main development libraries of the GNOME platform and include the graphical libraries, the component model, the accessibility libraries, Configuration and Lockdown, printing and the main GNOME desktop library, libGNOMEui.

Main Desktop modules. These include the main components of the GNOME desktop environment, such as file manager, panel, window manager and Evolution, GNOME's primary groupware and personal information manager, and the multimedia framework, Gstreamer.

Secondary desktop modules, which include secondary libraries and secondary elements of the GNOME desktop as well as the end-user productivity applications included in the GNOME release.

Development tools and processes. This area comprises both tools for development and tasks related to releases, quality control issues and unspecified bug fixes.

Peripheral activities which comprise non-coding aspects of contribution, such as documentation, translation, artwork, etc.

In terms of the distribution among volunteers and paid contributors, the data analysis indicates an almost even split: the sample consists of 98 volunteers and 101 affiliated individuals. Furthermore, as shown in Figure 6-1 the data allow for a more detailed categorization of paid developers based on whether they were employed to contribute exclusively to GNOME, to other F/OS projects, or both.

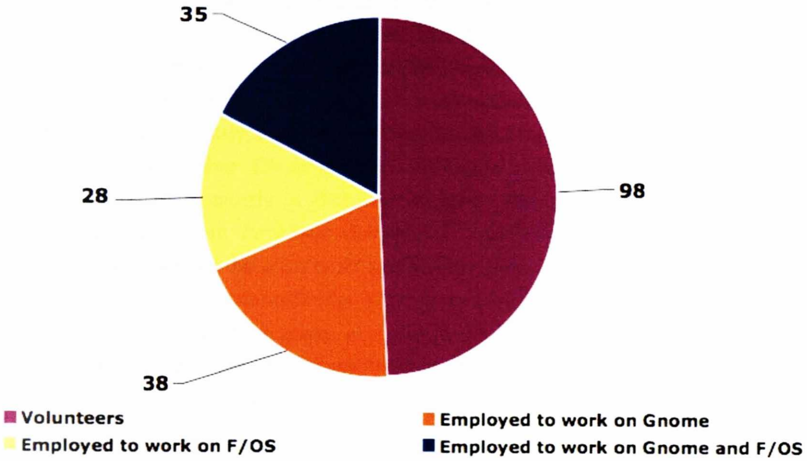


Figure 6-1: Breakdown of GNOME Foundation respondents according to their employment status (volunteers and affiliated)

Source: GNOME Foundation survey, N=199

The data indicate that[51] GNOME developers are sponsored by a variety of contributing organizations. The paid developers in the sample were hired by 41 institutions, including Small and Medium Enterprises (SMEs), large companies and research institutions. The three most important employers are Novell Inc. and RedHat Inc. with 16 developers each, and Sun Microsystems Inc. with 12 developers working on GNOME and F/OS developments. The two biggest corporations in the sample, IBM and Nokia, appear to participate only marginally in the project, at least at the level of the Foundation, with one contributor each. However, more detailed examination of developed products and services reveals some interesting patterns of cooperation among larger and smaller players. OpenedHand Ltd, an SME based in London, and Fluendo SL, an SME based in Barcelona, for example, have cooperated with Nokia on the development of its 770 Internet Tablet and the Maemo Platform. The sample also includes a considerable number of self employed developers (8) who undertake jobs on a subcontractor basis.

Table 6-1 presents a comparison between clusters of GNOME respondents based on the modules and activities they indicated participation in during their involvement in the project.[52] In the context of this study, this represents long-term involvement. The members grouped in *Cluster 1*, by far the largest cluster with 81 members, are mostly active in the areas of *Secondary Desktop* and *Development Tools and*

[51] For a detailed breakdown of the participating organizations and the number of developers they employ to work on GNOME see list Appendix (p. 252).

[52] The analysis was based on the complete linkage (furthest neighbour) method. The dendrogram can be found in the Appendix (pp. 253-254).

Processes. The most frequently cited area of contribution in *Secondary Desktop* is End-User Applications (56 contributors indicated that they had been contributors), and in *Development Tools and Processes*, developer-oriented tools, such as bug buddy, gedit, etc. were the most frequently cited.[53] *Cluster 2* encompasses a fairly small team of developers, predominantly focused on *Core/Platform Module* development, but also active in *Secondary Desktop*. *Cluster 3*, the second largest group, comprises respondents who reported activity mostly in *Peripheral Activities*. The most frequently reported area of contribution in *Peripheral Activities* is *Translation*, with 44 contributors indicating this as one of the areas of activity in the course of their involvement in the project.[54] *Cluster 4* groups contributors working mainly on *Main Desktop*. *Cluster 5*, the smallest cluster, comprises a group of contributors active in *Core/Platform Module* development and *Development Tools and Processes*. These are possibly contributors who, in addition to platform development, are also involved in releases and Quality Assurance (QA) issues.

Long-term involvement: Cluster analysis

CLUSTER	**Total of contribution in Core /Platform Modules**	**Total of contribution in Main Desktop**	**Total of contribution in Secondary Desktop/apps**	**Total of contribution in Development Tools and Processes**	**Total of contribution in Peripheral Activities**
1 **No of members:81**	Mean:0	0.0864	0.7654	0.5926	0.2222
2 **No of members:19**	Mean:1.3684	0.2632	0.4211	0	0
3 **No of members:66**	Mean:0.0909	0.0606	0.1667	0.1212	1.30366
4 **No of members:28**	Mean:0.1786	1.2143	0.2143	0.2857	0.2143
5 **No of members:5**	Mean:1.4	0.6	0.2	1.8	0.2

Table 6-1: Mean analysis of a five-cluster solution produced by hierarchical cluster analysis grouping individuals on the basis of their scores in each area of development

Source: GNOME Foundation survey, N=199

Note: These scores were generated by summing all the reported modules/activities for each area of development, i.e. an individual reporting activity in two different categories of core modules, would obtain a total score of 2 for that particular area of development. The numbers in the table represent group mean scores for contributions in each of the five areas of development

[53] See frequency tables of GNOME Foundation respondents' areas of contribution in the Appendix (p. 255-257).

[54] See frequency tables of GNOME Foundation respondents' areas of contribution in the Appendix (pp. 255-257).

In order to test whether the differences in the long-term patterns of overall involvement of volunteers and paid contributors are statistically significant, each individual was assigned a score based on the number and criticality of areas they reported involvement in. The scores were subjected to ANOVA (Analysis of Variance). Table 6-2 indicates the statistically significant relationships between the groups.

Long-term involvement: ANOVA[55]

	Paid to work on F/OS	Paid to work only on GNOME	Paid to work on GNOME and F/OS	Paid to work on GNOME and on GNOME & F/OS	All Groups of Affiliated
Volunteers	Not significant	***	***	***	***
Paid to work on F/OS		***	**	***	
Paid to work only on GNOME			**		

Table 6-2: Significance of long-term patterns of contribution between different groups of contributors

Source: GNOME Foundation survey, N=199

Note: *=significant at p<.001, **=significant at p<.01, *=significant at p<.05**

The analysis indicates that there are significant differences in the overall patterns of contribution between volunteers, and developers hired to contribute to GNOME either on an exclusive basis (Paid to work on GNOME) or to work also on other F/OS projects (Paid to work on GNOME and F/OS). The most significant difference was that between volunteers and 'Paid to work on GNOME, and Paid to work on GNOME and F/OS' (F=42.384, p<.0001). As can be seen from Table 6-2, however, there appear to be no statistically significant differences between volunteers and developers paid to work on F/OS, but not on GNOME (Paid to work on F/OS).

How do these findings relate to the primary patterns of contribution between developers and volunteers? Figure 6-2 depicts the breakdown of respondents' involvement according to their main area of contribution, that is, the module/activity they reported being most active in the six months prior to the survey. In this study this is their main contribution. What stands out is the overwhelming presence of paid developers in *Core/Platform* and the very strong presence of volunteers in *Secondary Desktop* and *Peripheral Activities*. More specifically, 85.7% (53.6%+32.1%) of all contributors involved in *Core Modules* are developers hired to work on GNOME, and more than 60% of the contributors primarily active in *Peripheral Activities* are

[55] For the precise F-values see Appendix (p. 258).

volunteers. In *Secondary Desktop*, 64.1% of the respondents are volunteers and in *Main Desktop*, more than 70% of developers are employed.

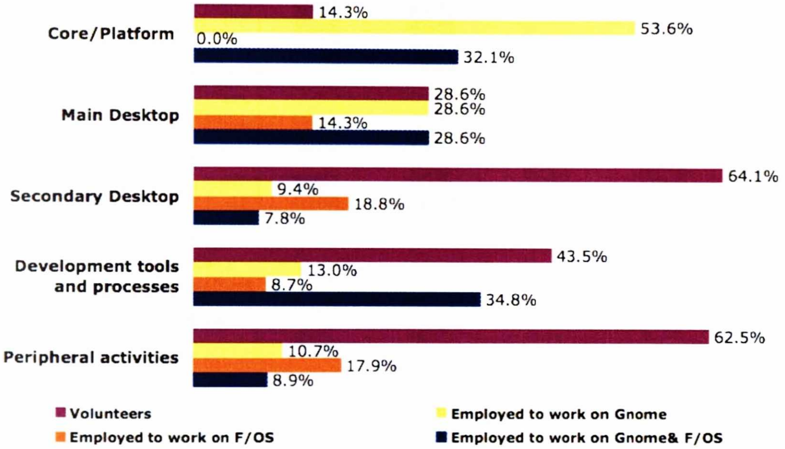
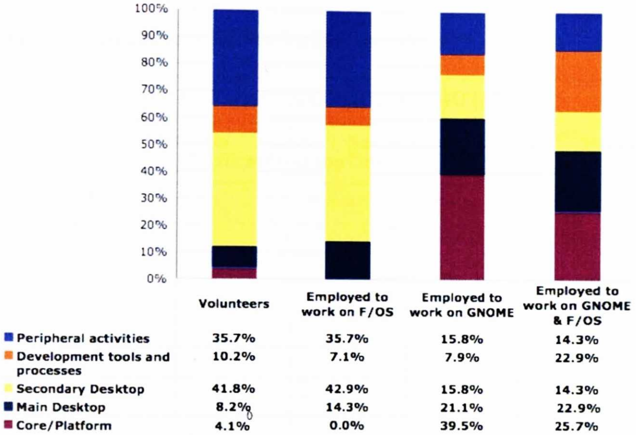


Figure 6-2: Analysis of involvement of paid and volunteer GNOME contributors in principal areas of development at the short-term level

Source: GNOME Foundation survey, N=199

Figure 6-3 depicts the distribution of each group's efforts in the five main areas of contribution. It is interesting to see that volunteers and contributors paid to work on F/OS, but not on GNOME, show very similar patterns of participation: they contribute almost equally intensively in *Secondary Desktop* (42%) and *Peripheral Activities* (35.7%). Similarly, dedicated GNOME hackers (Paid to work on GNOME and Paid to work on GNOME and F/OS) appear to have almost identical patterns of involvement in four out of the five areas of development *(Core/Platform, Main Desktop, Secondary Desktop* and *Peripheral Activities*).



	Volunteers	Employed to work on F/OS	Employed to work on GNOME	Employed to work on GNOME & F/OS
🟦 Peripheral activities	35.7%	35.7%	15.8%	14.3%
🟧 Development tools and processes	10.2%	7.1%	7.9%	22.9%
🟨 Secondary Desktop	41.8%	42.9%	15.8%	14.3%
⬛ Main Desktop	8.2%	14.3%	21.1%	22.9%
🟥 Core/Platform	4.1%	0.0%	39.5%	25.7%

Figure 6-3: Distribution of effort by group at the short-term level

Source: GNOME Foundation survey, N=199

To what extent are these differences statistically significant? Table 6-3 reports the results of the crosstabulation of volunteers and affiliated contributors based on data related to their main contributions. The crosstabulation has a chi-square value of χ^2 (df=8 and N=199)=53.372, p<0.0001) which indicates that there is a significant association between the selected groups and the specified areas of development. The adjusted residuals inform us that the most significant patterns of association are between volunteers and *Core/Platform* modules (-4), volunteers and *Main Desktop* modules (-2.4), volunteers and *Secondary Desktop* modules (2.9), volunteers and *Peripheral Activities* (2.3) employed to work on GNOME and GNOME and F/OS and *Core/Platform* modules (5.8) employed to work on GNOME and GNOME and F/OS and *Secondary Desktop* (-3.9) and *Peripheral Activities* (-3.1). According to these results, there is a larger proportion of paid developers than might be expected if the variables where independent that contribute to the *Core/Platform* and *Main Desktop* modules, while volunteers clearly contribute more to *Peripheral Activities* and to the area of *Secondary Desktop*. The only area of development that is significant for developers paid to work on F/OS is *Core/Platform* (-2.3). This means that there a negative association between paid developers employed to work on F/OS and this area of contribution.

Main Contribution by Affiliation/ CROSSTABULATION

		MAIN AREA OF CONTRIBUTION					
		Core/ Platform	Main Desktop	Secondary Desktop	Development tools and processes	Peripheral activities	Total
Volunteers	Count	4	8	41	10	35	98
	Expected Count	13.8	13.8	31.5	11.3	27.6	98
	% within the Group	4.1%	8.2%	41.8%	10.2%	35.7%	100.0%
	% within the Area	14.3%	28.6%	64.1%	43.5%	62.5%	49.2%
	Adjusted Residual	-4	-2.4	2.9	-0.6	2.3	
Paid to work on GNOME and GNOME & F/OS	Count	24	16	11	11	11	73
	Expected Count	10.3	10.3	23.5	8.4	20.5	73
	% within the Group	32.9%	21.9%	15.1%	15.1%	15.1%	100.0%
	% within the Area	85.7%	57.1%	17.2%	47.8%	19.6%	36.7%
	Adjusted Residual	5.8	2.4	-3.9	1.2	-3.1	
Paid to work on F/OS	Count	0	4	12	2	10	28
	Expected Count	3.9	3.9	9	3.2	7.9	28
	% within the Group	0.0%	14.3%	42.9%	7.1%	35.7%	100.0%
	% within the Area	0.0%	14.3%	18.8%	8.7%	17.9%	14.1%
	Adjusted Residual	-2.3	0	1.3	-0.8	1	

Table 6-3: Crosstabulation of main area of contribution* affiliated and volunteers

Source: GNOME Foundation survey, N=199

Table 6-4 shows the statistical significance of the patterns of contribution for various combinations of affiliated and volunteer workers. The results confirm the findings in Table 6-3, that is, that the most statistically significant differences in terms of areas of contribution are between volunteers and developers hired to work on GNOME.

Main Contribution: Detailed overview of differences between Volunteers and Affiliated[56]

Paid to work on F/OS	Paid to work only on GNOME	Paid to work on GNOME and F/OS	Paid to work on GNOME and GNOME & F/OS	All Paid

[56] For the corresponding table with detailed chi-square values see Appendix (p. 258).

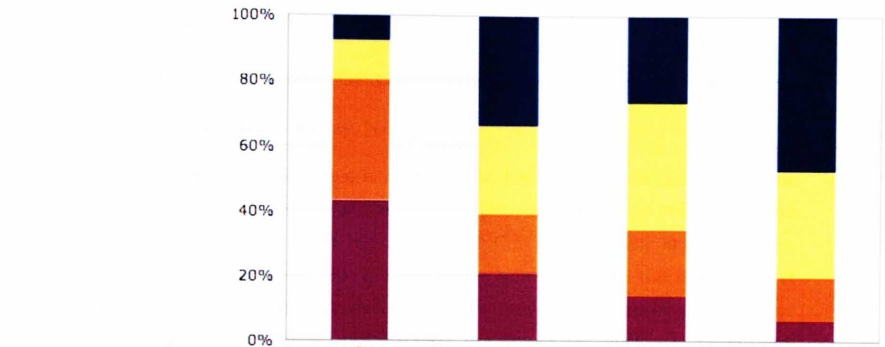
Volunteers	Not significant	Core/Platform*** Main Desktop** Secondary Desktop** Peripheral Activities*	Core/Platform*** Main Desktop* Secondary Desktop** Peripheral Activities*	Core/Platform*** Main Desktop* Secondary Desktop*** Peripheral Activities**	Core/Platform*** Main Desktop* Secondary Desktop** Peripheral Activities*
Paid to work on F/OS		Core/Platform*** Secondary Desktop*	Core/Platform** Secondary Desktop* Peripheral Activities*	Core/Platform** Secondary Desktop** Peripheral Activities*	
Paid to work only on GNOME			Not significant		

Table 6-4: Detailed overview of patterns of contribution between volunteers and affiliated at the short-term level for different group combinations

Note: *=significant at p<.001, **=significant at p<.01, *=significant at p<.05**

How do these patterns of contribution relate to the more general, long-term patterns of involvement as indicated by the cluster analysis? The two sets of findings seem to be consistent in the sense that both point to: a) a fairly small group of core developers focusing on *Platform* and *Main Desktop* development (Clusters 2 and 4); and b) two substantial groups of individuals involved in *Secondary Desktop* (Cluster 1) and *Peripheral Activities* (Cluster 3). This could indicate that most people tend to be consistent throughout their career in the project in contributing to the same, broadly defined, areas of development. At the same time analysis of the data on the main patterns of contribution indicates that the group of core developers consists primarily of paid developers and that the people involved in more user-oriented and peripheral aspects of development are predominantly volunteers.

How do these findings relate to event attendance? Figure 6-4 indicates how frequently volunteers and paid developers participated in GUADEC, GNOME's leading community event. As we can see, close to 65% (38.2%+26.3%) of developers employed to work on GNOME participated in all three GUADEC conferences between 2003-2005 whereas 66% of volunteers and 57% of those contributors hired to work on F/OS but not on GNOME did not participated in any of these events. The chi-square statistic has a value of χ^2 (df=6 and N=199)=51.425, p.<0.0001, indicating that the relationship between affiliation and conference attendance is indeed significant. The adjusted residuals in Table 6-5 which includes the results of the crosstabulation of events and affiliation, allows us to identify the most significant patterns of attendance.



	0	1	2	3
■ Employed to work on Gnome & F/OS	11.8%	26.5%	23.5%	38.2%
■ Employed to work on Gnome	18.4%	21.1%	34.2%	26.3%
■ Employed to work on F/OS	57.1%	14.3%	17.9%	10.7%
■ Volunteers	66.3%	16.3%	12.2%	5.1%

Figure 6-4: Frequency of participation of volunteers and affiliated in GUADECs held between 2003-05

Source: GNOME Foundation survey, N=199

As can be seen there is a large proportion of volunteers who did not attend any events between 2003-2005 (5.5) and a very small proportion who attended all three events (-4). By contrast, the adjusted residuals for paid developers suggest that the majority of the members of this group rarely miss a major community event (4.7 for attendance at 3 GUADECs and -6.5 for non-attendance).

GUADEC attendance by Affiliation / CROSSTABULATION

		Number of attended GUADECS (2003-2005)			
		0	**1**	**2**	**3**
Volunteers	Count	65	16	12	5
	% within the Group	0.663	0.163	0.122	0.051
	% within the Event	0.699	0.432	0.316	0.161
	Adjusted Residual	5.5	-0.8	-2.4	-4
Paid to work on GNOME and GNOME & F/OS	Count	12	17	21	23
	% within the Group	0.164	0.233	0.288	0.315
	% within the Event	0.129	0.459	0.553	0.742
	Adjusted Residual	-6.5	1.3	2.6	4.7
Paid to work on F/OS	Count	16	4	5	3
	% within the Group	0.571	0.143	0.179	0.107
	% within the no of Events	0.172	0.108	0.132	0.097

Adjusted Residual	1.2	-0.6	-0.2	-0.8

Table 6-5: Crosstabulation of no of GUADECs attended by affiliated and volunteers

Source: GNOME Foundation survey, N=199

Table 6-6 presents the significance levels for event attendance for the different group combinations. One of the clearest patterns relates to volunteers and hackers employed to work on F/OS but not on GNOME, where the analysis shows that the differences between the two groups is not significant. At the same time it confirms that the most significant differences are among volunteers and hackers employed to work on GNOME, and hackers employed to work on F/OS, and GNOME dedicated hackers.

GUADEC Attendance: Detailed overview of differences between Affiliated and Volunteers[57]

	Paid to work on F/OS	**Paid to work only on GNOME**	**Paid to work on GNOME and F/OS**	**Paid to work on GNOME and GNOME & F/OS**	**All Paid**
Volunteers	Not significant	***	***	***	***
Paid to work on F/OS		***	Not significant	***	
Paid to work only on GNOME			Not significant		

Table 6-6: Detailed overview of patterns of conference attendance between volunteers and affiliated for different group combinations. Source: GNOME Foundation survey, N=199

Note:*=significant at p<.001, **=significant at p.<0.01, *=significant at p<.05**

In order to examine the relationship between affiliation, area of contribution and GUADEC attendance more deeply, several logistic regression models were built and tested. These included models predicting whether a contributor is a volunteer or is affiliated (belonging to one of the three groups of paid developers) or whether a contributor is a volunteer or a GNOME dedicated developer (Paid to work on GNOME or GNOME and F/OS). The models used different combinations of the following variables: main area of contribution, level of responsibility, whether a maintainer or not,[58][59] number of modules maintained and GUADEC attendance.

[57] For the corresponding table with detailed chi-square values see Appendix (p. 259).

[58] The analysis was based on the ENTER method since stepwise techniques are prone to be influenced by random variations in the data, which can have a negative effect on the replicability of the results within the same sample (Studenmund and Cassidy, 1987). Analysis of the residuals and tests for collinearity were conducted. The results were within acceptable parameters as specified by (Field, 2005).

[59] This relationship is examined more extensively in the next section.

The logististic regression model, whose parameters are presented in Table 6-7 predicts whether volunteer or GNOME dedicated paid developer status on the basis of the main area of contribution, number of module(s) maintained and GUADEC attendance. This model was chosen because it provides a better overall fit for the data than a model that uses the same dependent variables, but predicts volunteer or generally affiliated developer status. The Wald statistic provides the information that in addition to GUADEC attendance (significant at $p<0.001$), the variables that are most significant in predicting whether a contributor is a volunteer or was hired to work on the GNOME project are contribution to *Core/Platform* modules (significant at $p<0.001$), *Main Desktop* modules (significant at $p<0.05$) and *Development Tools and Processes* (significant at $p<0.05$).

The variables that have more predictive value in assessing whether an individual is a dedicated GNOME paid developer are: contribution in *Core/Platform* modules (B=2.673), contribution in *Main Desktop* modules (B=1.603), contribution in *Development Tools and Processes* (B=1.403) and GUADEC attendance (B=1.046). According to the model a developer who attended no GUADECs, but who contributes in Core/Platforms is 14.486 times more likely to be a paid developer than someone who involved in a *Peripheral Activity*. A developer that attended all three GUADECs and contributed to the *Core/Platform* and *Main Desktop* modules is 22 times (2.847+14.486+4.97) more likely to be a GNOME dedicated developer than an individual involved in *Peripheral Activities* who did not attend any GUADECs.[60].This model improves the chances by 25.9% of correctly predicting whether a contributor is affiliated or is a volunteer.[61]

Logistic Regression: GNOME respondents

	B	Wald	Sig.	Exp(B)
Main area of contribution		22.623	0	
Core/Platform(1)	2.673	11.88	0.001	14.486
Main Desktop(2)	1.603	5.847	0.016	4.97
Secondary Desktop(3)	-0.356	0.372	0.542	0.701
Development tools and processes(4)	1.403	4.676	0.031	4.067
No of Modules maintained	0.226	0.654	0.419	1.254
No of attended GUADECs	1.046	26.873	0.000	2.847
Constant	-2.377	23.78	0	0.093

Table 6-7: Model Parameters for Logistic Regression. Source: GNOME Foundation survey, N=170

[60] It is interesting to note that the model that includes the long-term view of engagement (based on scale of contribution) rather than the short-term view afforded by the Main area of Contribution predicts affiliation almost as well as the model presented here, a result that is consistent with the previous suggestion that contributors tend not to move between different areas of contribution.

[61] That is, compared to the baseline model which predicts that a developer is always going to be affiliated.

6.3.2 *The GNOME maintainer network*

The data collected for the study of the GNOME maintainer network include 110 modules,[62] which at the time of the 2.10.0 GNOME release were maintained by 92 individuals. The majority of the modules (76.4%) were maintained by a single individual; a significant number of modules (21%) were maintained by two individuals and a small number of critical modules[63] (4.5%) were maintained by groups of three or four developers. Seen from the perspective of individual ownership, this was distributed among a large number (76) of developers who were responsible either for one (64 developers) or two modules (12 developers), and a smaller group of 16 individuals who maintained more than two modules. As will be demonstrated later these were predominantly paid developers. Figure 6-5 provides a more detailed distribution of GNOME maintainers based on the number of modules they maintain.[64] 'Maintainership' is used to refer to a maintainer, which, although in some cases this can be the case, generally in F/OS means more than passively maintaining code; maintainers in F/OS often act as gatekeepers in terms of choosing which patches they allow to be incorporated in the main development tree and dictating the project's development policy.[65]

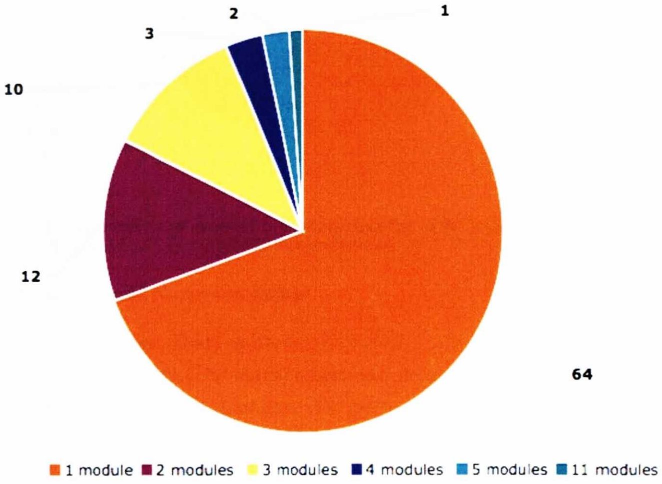


Figure 6-5: Distribution of GNOME contributors according to number of modules maintained.

[62] For a discussion of the methodology devised and the decisions about what to include in the maintainer network and the protocols implemented to identify the maintainers and their affiliations, see Chapter 3, section 3.8.

[63] These are Glib, Nautilus, Libwnck, start-up notification, and Gal.

[64] This observation is supported by the cluster analysis of GNOME maintainers, whose results are not presented here, but which pointed to a small group of exceptionally active maintainers. For more information see dendrogram and means analysis in the Appendix (pp. 260-261 and p. 264).

[65] For a detailed explanation of what the role of the maintainer implies, see Chapter 3, section 3.8.

Source: Online search, N=92 (maintainers)

Similar to the GNOME Foundation itself, the GNOME maintainer network is characterized by an almost even split between affiliated and non-affiliated developers.[66] More specifically, the 2.10.0 release was accomplished by 42 non-affiliated and 50 affiliated maintainers. Figure 6-6 presents the distribution of affiliated and non-affiliated developer maintainership. As can be seen 67 modules (61%) were maintained exclusively by affiliated developers, 36 (33%) were maintained solely by non-affiliated developers and 7 modules (6%) were maintained by affiliated and non-affiliated programmer cooperation.

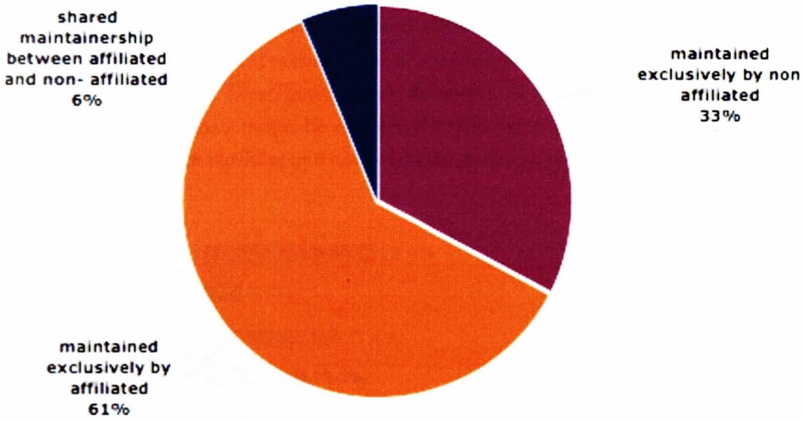


Figure 6-6: Distribution of modules maintained by affiliated, non-affiliated and by affiliated and non-affiliated GNOME developers

Source: Online search, N=110 (modules)

The companies that sponsor GNOME maintainers are a subset of organizations that sponsor GNOME Foundation members. Among the 11 companies that have hired GNOME maintainers, the main employers are Novell Inc. with 23 employed maintainers, RedHat Inc. with 9 and Sun Microsystems Inc. with 7.

As can be seen from Figure 6-7 most of the *Core/Platform* and *Main Desktop* modules are maintained exclusively by affiliated developers. This is consistent with the results of the analysis of the GNOME Foundation members' network. In contrast to the findings for Foundation members, however, where the area of *Secondary Desktop* (which included Applications) was mostly developed by volunteers, maintainership in this area is divided almost equally between non-affiliated and affiliated developers.

[66] In terms of the overlap between the two networks there were 27 maintainers who were not members of the GNOME Foundation and 42 who were not among the Foundation respondents.

At the same time, non-affiliated developers appear to maintain more *Peripheral* modules, such as those related to developer tools and documentation (grouped under "Other").

Analysis of the significance of the relationship between maintainership and affiliation concentrates on cases of exclusive maintainership.[67] The chi Square test for the crosstabulation (see Table 6-8) of cases of exclusive maintainership with the main areas of the code base has the value χ^2(df=4 and N=103)=12.071, p<0.05. This means that there is an association between the areas that developers maintain and their affiliation (or lack thereof). A closer look at each area of the code base reveals,[68] however, that the relations that are statistically significant are in fact only those for *Core/Platform* (adjusted residuals=-2,2) and *Other (Peripheral)* module*s* (adjusted residuals=-2.4).[69] This indicates that there is a significantly larger proportion of paid developers maintaining *Core/Platform* and volunteer developers maintaining *Other (Peripheral)* modules than might be expected. Affiliated developers clearly prefer to maintain *Core/Platform* modules and non-affiliated prefer to maintain *Other (Peripheral)* modules.

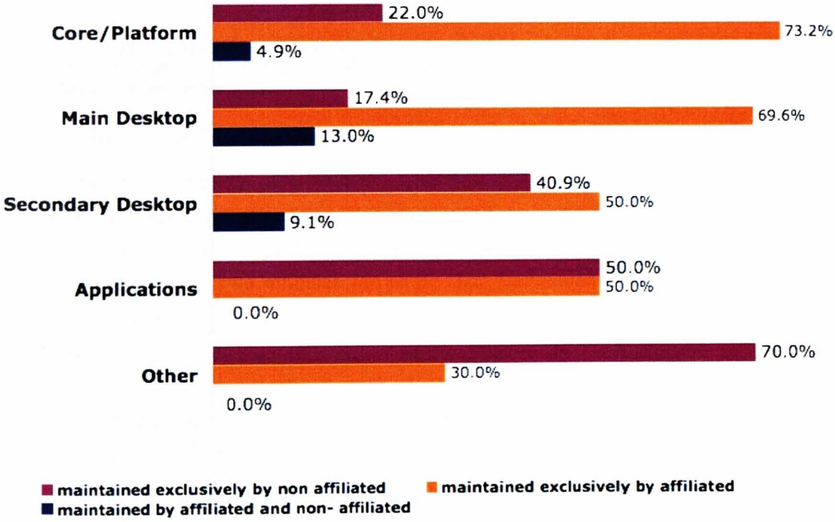


Figure 6-7: Analysis of modules maintained by affiliated and non-affiliated GNOME contributors in principal areas of development.

Source: Online search, N=110 (modules)

[67] This is necessary since taking into account all 3 categories of maintainership resulted in too high a percentage of expected frequencies under 5, which results in a loss of statistical power. For an explanation see Field (2005).

[68] As also supported by the ANOVA analysis for each area of the code base (see Appendix p 262).

[69] This is confirmed by Crosstabulation in the (Appendix p. 263).

Maintainership Area by Affiliation / CROSSTABULATION

		Area of Module Maintained					
		Core/Platform	Main Desktop	Secondary Desktop	Applications	Other	Total
Maintained exclusively by non-affiliated	Count	9	4	9	7	7	36
	Expected Count	13.6	7	7	4.9	3.5	36
	% within the Group	25.0%	11.1%	25.0%	19.4%	19.4%	100.0%
	% within the Area	23.1%	20.0%	45.0%	50.0%	70.0%	35.0%
	Adjusted Residual	-2	-1.6	1	1.3	2.4	
Maintained exclusively by affiliated	Count	30	16	11	7	3	67
	Expected Count	25.4	13	13	9.1	6.5	67
	% within the Group	44.8%	23.9%	16.4%	10.4%	4.5%	100.0%
	% within the Area	76.9%	80.0%	55.0%	50.0%	30.0%	65.0%
	Adjusted Residual	2	1.6	-1	-1.3	-2.4	

Table 6-8: Crosstabulation of cases of exclusive maintainership (modules maintained exclusively by non-affiliated and modules maintained exclusively by affilitated) with the areas the modules belong in.

Source: Online search, N=103 (modules)

Another important relationship is that between affiliated and non-affiliated developers and the numbers of modules maintained. In order to determine whether this relation is statistically significant, maintainers were divided into two groups: those that maintained up to two modules and those that maintained more than two. The association is significant at p<.05 (χ^2 (df=1, N=92)= 5.650), which indicates that the two variables are dependent. The odds ratios, which measure the probability of an event occurring and which are calculated on the basis of the counts included in Table 6-9 indicate that an affiliated developer is 4.5 times more likely to maintain more than two modules as a non-affiliated developer.

No of modules maintained by Affiliation/ CROSSTABULATION

		Group Category according to no of modules maintained		
		Maintaining up to 2 modules	Maintaining more than 2 modules	Total
Non-affiliated	Count	39	3	42
	Expected Count	34.7	7.3	42
	% within each Group	92.9%	7.1%	100.0%
	% within each Group category	51.3%	18.8%	45.7%

	Adjusted Residual	2.4	-2.4	
Affiliated	Count	37	13	50
	Expected Count	41.3	8.7	50
	% within Volunteer vs. affiliated	74.0%	26.0%	100.0%
	% within each Group Category	48.7%	81.3%	54.3%
	Adjusted Residual	-2.4	2.4	

Table 6-9: Crosstabulation of affiliation with number of modules maintained. Source: Online search, N=92 (maintainers)

The findings for patterns of affiliated and non-affiliated developer maintainership are consistent with the findings on patterns of contribution in the GNOME Foundation network. Developers affiliated to an organization involved in F/OS development maintain modules associated with the most critical aspects of the project's development, *Core/Platform* and *Main Desktop* modules. Non-affiliated developers are mostly involved in maintenance of *Peripheral* modules, that is, parts of the code base that are associated with *Development Tools*. Paid developers also maintain more modules than do non-affiliated contributors. Lastly, the absence of cooperatively maintained modules supports the hypothesis that the two groups – affiliated and non-affiliated developers - are quite distinctive.

Based on the results of the analysis for the KDE project, the following can be stated for the hypotheses formulated at the beginning of this chapter:

1. Paid developers are more likely to contribute to critical parts of the code base. **CONFIRMED**
2. Paid developers are more likely to maintain critical parts of the code base. **CONFIRMED**
3. Volunteer contributors are more likely to participate in aspects of the project that are more geared towards the end-user. **CONFIRMED**
4. Programmers and peripheral contributors are not likely to participate equally in major community events. **CONFIRMED**

6.4 The KDE community

The results for the KDE project are different from those obtained for GNOME. In GNOME the patterns of contribution and maintainership were similar, but in KDE they are quite different. In KDE e.V., there are almost no statistically significant differences in main and long-term patterns of contribution between volunteers and paid developers. At a descriptive level, volunteers reported being more active than paid developers in *Core/Platform* and *Main Desktop* modules. Examination of the KDE maintainer network, however, reveals important differences in maintainership patterns between affiliated and non-affiliated contributors. Due to the restrictions created by the considerably smaller pool of respondents in the case of the KDE e.V. network the analysis is more limited.

6.4.1 The KDE e.V. network

As indicated in Chapter 3, the KDE e.V. members' survey had a 55.7% response rate (63 from 113 KDE e.V. members registered in April 2005). The analysis and presentation of the findings in this section is structured similarly to the analysis of the GNOME Foundation. However, where the number of responses in GNOME allowed tests for differences between volunteers and different types of paid contributors, the KDE analysis is limited to testing for differences between volunteers and affiliated contributors generally, and volunteers and contributors paid to work specifically on KDE.

Below the five main areas of development[70] that form the basis of the analysis are described:

> **Core/Platform modules** comprise the main development libraries (KDE development environment) the component model (KParts), Configuration and Lockdown (Kiosk) and the main libraries of the KDE desktop environment that are integral to platform and desktop development, such as KDE Core, KDESU, KDEeui.
>
> **Main Desktop modules** include the main components of the KDE desktop, that is most of modules in release 3.4 included in the KDE Base that are considered to belong to KDE proper such as file manager, panel, window manager, KDE, and KDE PIM, KDE's primary groupware and personal information manager, and the multimedia framework.
>
> **Secondary desktop modules** include KDE's secondary desktop components and the user productivity applications included in the release, including KOffice.
>
> **Development tools and processes** includes tools for development as well as tasks related to releases, quality control issues, and unspecified bug fixes.
>
> **Peripheral** activities comprise non-coding type contributions, such as documentation, translation, artwork, etc.

As in the case of GNOME, the KDE e.V. members who responded to the survey were almost equally divided between volunteers (34) and affiliated developers (29). As Figure 6-8 indicates of the 29 paid developers, 19 were employed to develop parts of the code base directly related to KDE, 5 were employed to hack on F/OS and 5 were employed to work on KDE and other F/OS software.

[70] For a more detailed account of the coding scheme see Chapter 3 Table 3.4.

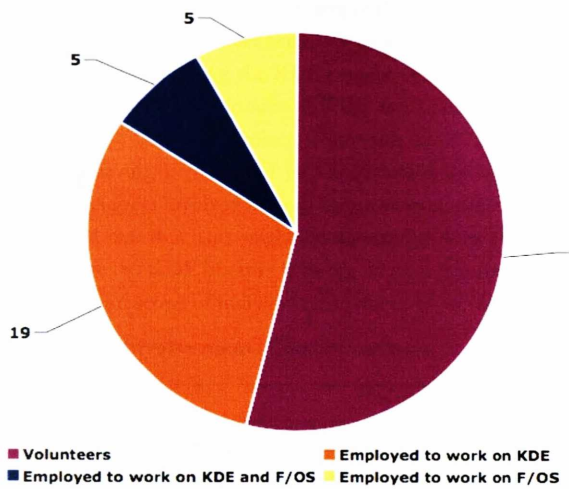


Figure 6-8: Breakdown of KDE e.V. respondents according to their employment status

Source: KDE e.V. survey, N=63

Perhaps in line with the more limited respondent pool, the number of organizations employing KDE e.V. respondents[71] is smaller than for the GNOME Foundation. Overall there are 16 organizations employing KDE e.V. programmers. The most important employers are Novell Inc. with five programmers, Trolltech ASA and Klarälvdalens Datakonsult AB each with three programmers. As in the case of GNOME, KDE e.V. also includes a group of self-employed programmers (4), who reported being subcontracted regularly to work on KDE.

The means comparisons of the four cluster solution[72] in Table 6-10 reveals the following distinctive groups of contributors: Cluster 1, which includes the largest number of respondents, represents a group of contributors who divide their time between contributing in the areas of the Main and Secondary Desktop, which in the case of KDE, includes predominantly end-user productivity applications. Cluster 2, the second biggest grouping of contributors with 16 members, includes developers working in the area of Core/Platform and Main Desktop. Cluster 3 represents a team of developers focused almost exclusively on Core/Platform development, and Cluster 4 includes the peripheral contributors. Unlike GNOME, no KDE e.V. respondent reported activity in *Translations*. The most frequently reported peripheral activity

[71] For the complete list of companies sponsoring KDE e.V. see Appendix (p. 265).

[72] As in the case of GNOME, the analysis adopted the furthest neighbour (complete linkage) method. The dendrogram can be found in the Appendix (p. 266). It is interesting to note that compared to the dendrogram of GNOME Foundation members where the distances between the clusters are relatively small, the clusters in the dendrogram for KDE e.V. members are more pronounced, i.e. the distances between them are larger. This indicates that members of KDE e.V. belong to more clearly distinguishable groups than the members of the GNOME Foundation.

among KDE e.V. members, with 11 respondents indicating participation, was Promotion/Advocacy. This includes activities such as helping with the organization of community events, representing the KDE project in Linux forums and other project conferences, helping with local groups of KDE and Linux users, etc. It is interesting that compared to the GNOME project, where only 24 of 199 contributors (Clusters 2 and 5) reported having contributed to *Core/Platform* modules, KDE has a larger proportion of developers involved in *Core/Platform* development (25 out of 63). A KDE developer pointed out that this might be associated with the fact that submitting incremental changes to KDE libraries is easier. To test this probability additional data regarding the size and scope of individual contributions would be required.

Long-term involvement: Cluster analysis

Complete Linkage	Total of contribution in Core/Platform	Total of contribution in Main Desktop	Total of contribution in Secondary Desktop	Total of contribution in Development tools & processes	Total of contribution in Peripheral Activities
1 **No of members: 27**	Mean: 0.11	0.44	0.56	0.3	0.07
2 **No of members: 16**	Mean: 1.06	1.44	0.63	0	0.19
3 **No of members: 9**	Mean: 1.56	0.22	0	0.22	0.22
4 **No of members: 11**	Mean: 0	0	0	0	1.45

Table 6-10: Mean analysis of a four-cluster solution produced by a hierarchical cluster analysis grouping individuals on the basis of their scores in each area of development

Note: These scores were generated by summing the reported modules/activities for each area of development, i.e. an individual that was reported to have been active in two different categories of core modules, would obtain a total score of 2 for that particular area of development. The numbers in the Table represent group mean scores for contributions in each area

Source: KDE e.V. survey, N=63

As indicated by the test statistics in Table 6-11 there appear to be no significant differences in the long-term patterns of volunteers' and paid developers' contributions or between volunteers and developers hired to work on an aspect of KDE.

Long-term view of involvement: ANOVA

	Paid to work on F/OS	Paid to work only on KDE	Paid to work on KDE and F/OS	Paid to work on KDE and KDE & F/OS	All Paid
Volunteers	Cannot be estimated	Not significant	Cannot be estimated	Not significant	Not significant
Paid to work on F/OS		Cannot be estimated	Cannot be estimated	Cannot be estimated	
Paid to work only on KDE			Cannot be estimated		

Table 6-11: Significance of long-term patterns of contribution between different groups of contributors

Note: ***=significant at p<.001, **=significant at p<.01, *=significant at p<.0

The long-term view of involvement is consistent with the short-term view as reflected in the areas respondents indicated being most active in (main contribution) in the six months prior to the survey. Figure 6-9 provides an overview of reported primary areas of contribution for volunteer and paid developers. As can be seen, more volunteer than paid developers were reported as being active in *Core/Platform* and *Main Desktop*, the two most critical areas of KDE development. In addition, volunteers are shown to be active in *Peripheral Activities*. On the other hand, paid developers appear to monopolize the area of *Development Tools and Processes*, which includes activities such as integration and porting and the development of applications such as Quanta Plus. However, as can be seen from Table 6-12 and Table 6-13 none of these differences is statistically significant.

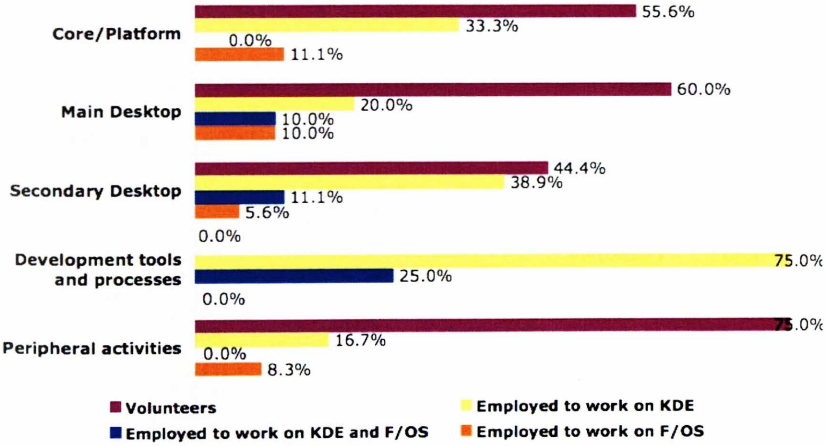
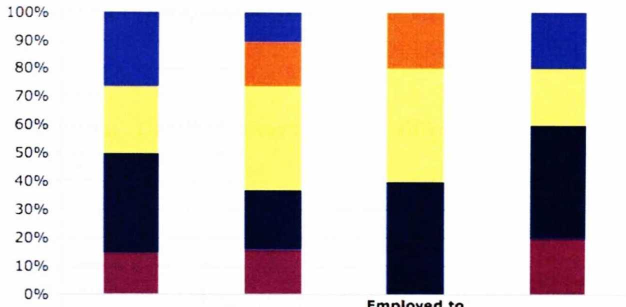


Figure 6-9: Analysis of involvement of paid and volunteer contributors in principal areas of development in the short-term

Source: KDE e.V. survey, N=63



	Volunteers	Employed to work on KDE	Employed to work on KDE and F/OS	Employed to work on F/OS
■ Peripheral activities	26.5%	10.5%	0.0%	20.0%
■ Development tools and processes	0.0%	15.8%	20.0%	0.0%
■ Secondary Desktop	23.5%	36.8%	40.0%	20.0%
■ Main Desktop	35.3%	21.1%	40.0%	40.0%
■ Core/Platform	14.7%	15.8%	0.0%	20.0%

Figure 6-10: Distribution of effort by group at the short-term level

Source: KDE e.V. survey, N=63

Main contribution by Affiliation/ CROSSTABULATION

Main Area of Contribution (recoded)[73]

		Primary modules (Core and MainDesktop)	Secondary Modules- (Secondary Desktop-apps)	Tertiary Modules (Development tools and proceses & Peripheral Activities)	Total
Volunteers	Count	16	9	9	34
	Expected Count	15.1	10.3	8.6	34
	% within the Group	47.1%	26.5%	26.5%	100.0%
	% within the Area	57.1%	47.4%	56.3%	54.0%
	Adjusted Residual	0.5	-0.7	0.2	
Affiliated	Count	12	10	7	29
	Expected Count	12.9	8.7	7.4	29
	% within the Group	41.4%	34.5%	24.1%	100.0%
	% within the Area	42.9%	52.6%	43.8%	46.0%
	Adjusted Residual	-0.5	0.7	-0.2	

[73] In order to test the statistical significance of patterns of contribution between volunteers and affiliated developers the five areas of development were recoded in the following way: Core/Platform and Main Desktop Modules were joined to form the 'Primary Area of Development', Secondary Desktop remained as was and the areas of Development tools and Processes and Peripheral Activities were joined to form the 'Tertiary Area of Development'. This recoding was necessary since taking into account all five areas of development resulted in an unacceptable loss of statistical power.

Table 6-12: Crosstabulation of recoded Main area of Contribution* affiliated and volunteers

Source: KDE e.V. survey, N=63

Main Contribution: Detailed overview of differences between Volunteers and Affiliated

	Paid to work on F/OS	Paid to work only on KDE	Paid to work on KDE and F/OS	Paid to work on KDE and KDE & F/OS	All Paid
Volunteers	Cannot be estimated	Not significant	Cannot be estimated	Not significant	Not significant
Paid to work on F/OS		Cannot be estimated	Cannot be estimated	Cannot be estimated	
Paid to work only on KDE			Cannot be estimated		

Table 6-13: Detailed overview of patterns of contribution between volunteers and affiliated at the short-term level for different group combinations

Note: *=significant at p<.001, **=significant at p<.01, *=significant at p<.05**

The apparent lack of differences between affiliated and volunteer developers does not apply if we examine conference attendance patterns. As Figure 6-11 shows paid developers tend to attend more conferences than volunteers: one in two (52%) developers paid to work exclusively on KDE had attended all three KDE conferences organized between 2003-2005. Only one in five volunteers (20.6%) had a similar record of attendance. The analysis indicates that these patterns are statistical significant for volunteers and the group of paid developers consisting of those who are paid to work on KDE and on KDE and F/OS (χ^2 (df=3, N=59)=8.427, p<0.5). More specifically, the odds ratio calculated by the counts included in Table 6-14 that a KDE dedicated developer is 2.29 times more likely than a volunteer to have attended all three KDE events.

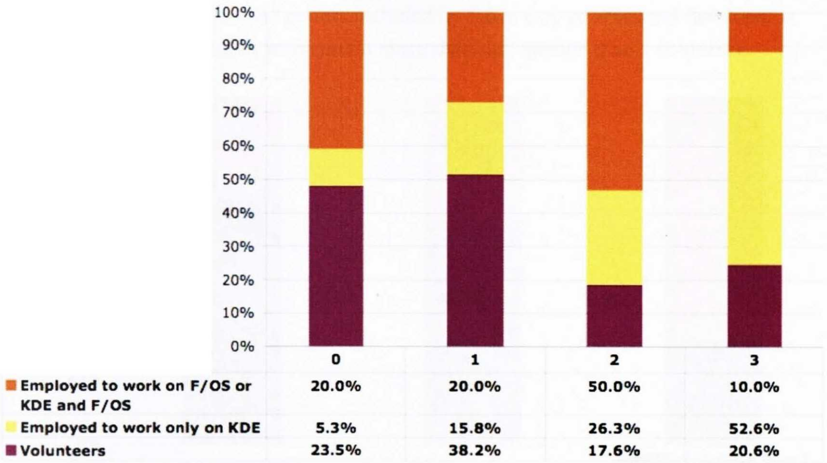


Figure 6-11: Frequency of participation in Akademy for volunteers and affiliated

Source: KDE e.V. survey, N=63

Akademy attendance by affiliation/ CROSSTABULATION

		Number of attended Akademies (2003-2005)				
		0	1	2	3	**Total**
Volunteers	Count	9	13	6	7	35
	Expected Count	6.5	10.1	7.7	10.7	35
	% within the Group	25.7%	37.1%	17.1%	20.0%	100.0%
	% within the Event	81.8%	76.5%	46.2%	38.9%	59.3%
	Adjusted Residual	1.7	1.7	-1.1	-2.1	
Employed to work on KDE and KDE and F/OS	Count	2	4	7	11	24
	Expected Count	4.5	6.9	5.3	7.3	24
	% within the Group	8.3%	16.7%	29.2%	45.8%	100.0%
	% within the Event	18.2%	23.5%	53.8%	61.1%	40.7%
	Adjusted Residual	-1.7	-1.7	1.1	2.1	

Table 6-14: Crosstabulation of no of Akademy events attended by volunteers and affiliated

Source: KDE e.V. survey, N=59

Another statistically important difference between volunteers and affiliated developers can be seen in the number of modules maintained (χ^2 (df=2, N=48)=7.113, p<.05). Figure 6-12 shows that whereas the numbers of volunteer and affiliated programmers who maintain only one module is almost the same, there is a considerable larger number of paid developers who maintain more than one module.

In particular, according to the counts included in Table 6-15 an affiliated developer is almost 3.75 times more likely to maintain more than one module than a volunteer.

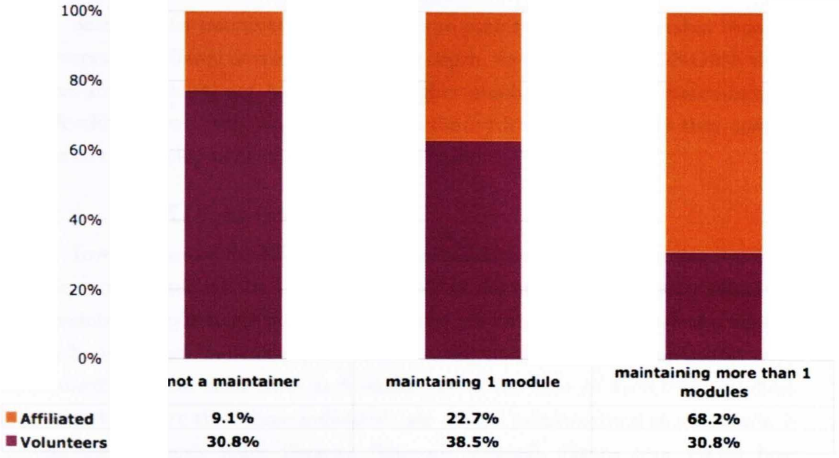


Figure 6-12: Distribution of number of modules maintained by volunteer and affiliated members of KDE e.V

Source: online search, N=48

No of modules maintained by Affiliation/ CROSSTABULATION

		Group Category according to no of modules maintained			
		Not a maintainer	Maintaining 1 module	Maintaining more than 1 modules	Total
Volunteers	Count	8	10	8	26
	Expected Count	5.4	8.1	12.5	26
	% within the Group	30.8%	38.5%	30.8%	100.0%
	% within the Group Category	80.0%	66.7%	34.8%	54.2%
	Adjusted Residual	1.8	1.2	-2.6	
Affiliated	Count	2	5	15	22
	Expected Count	4.6	6.9	10.5	22
	% within the Group	9.1%	22.7%	68.2%	100.0%
	% within the Group Category	20.0%	33.3%	65.2%	45.8%
	Adjusted Residual	-1.8	-1.2	2.6	

Table 6-15: Crosstabulation of affiliation with number of modules maintained

Source: Online search. N=48

Section 6.4.2 examines the differences in patterns of maintainership between volunteers and affiliated developers in greater depth. As in the case of GNOME these patterns are associated not only with the higher number of modules maintained by paid developers, but most importantly, with the criticality of modules they control relative to the development of the entire code base.

6.4.2 *The KDE maintainer network*

Investigation of the KDE maintainer network involved 111 maintainers and 191 modules which made up the 3.4 release. 84.8% of the modules/subprojects examined were maintained by a single individual. In only 7.3% of cases was control of a module shared between two individuals, and a very small number of modules (2.6%) were maintained by more than three individuals. As in the case of GNOME, modules maintained by more than three individuals are critical infrastructural elements which include the project's main libraries (kdecore, kdeeui). Figure 6-13 shows how maintainership is distributed among individuals: the majority of the developers (60.4%) maintained only one module. In addition, a relatively high number of programmers (18.9%) maintained two modules. Compared to GNOME, however, the KDE network has a noticeably higher number of individuals (23 vs 16) who, at the time of the study, were maintaining more than three modules. The smaller number of cases of cooperative maintainership and the larger number of cases of maintainers in charge of more than three modules, might be an indication of a higher concentration of maintainership in KDE than in GNOME. In order to support this observation, however, further evidence would be needed, for example, from an examination of the way releases are managed and the policies for accepting patches, which vary widely from one maintainer to another.

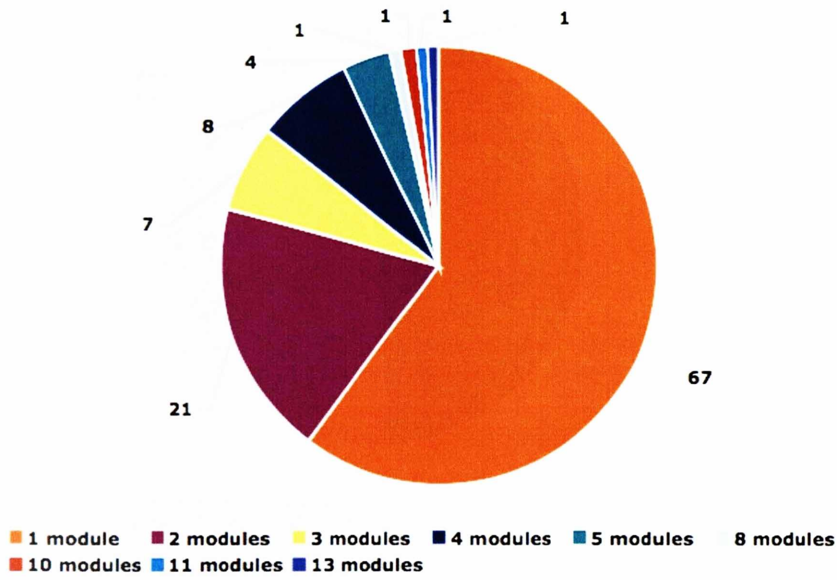


Figure 6-13: Distribution of KDE contributors according to number of modules maintained

Source: Online search, N=111 (maintainers)

The KDE e.V. maintainer network includes 72 non-affiliated and 39 affiliated developers.[74] This is a considerable contrast to GNOME which is characterized by an almost even split between volunteers and paid programmers. However, like GNOME, cases of affiliated and non-affiliated maintainers sharing custody of the same module are very few. As shown in Figure 6-14 only 3% of the modules examined were jointly maintained by affiliated and non-affiliated programmers.

KDE maintainers are employed by a score of companies mostly the same ones that employ KDE e.V. members.[75] As in the case of the KDE e.V. network the most important employers are Novell, Inc. with eight developers, Trolltech ASA with five developers and Klarälvdalens Datakonsult AB also with five developers.

[74] 82 of the 111 KDE maintainers were not among the KDE e.V. survey respondents and 67 of them were not members of KDE e.V. For a discussion on the protocol used to identify affiliation see Chapter 3, section 3.8.

[75] A full list of the companies and the number of maintainers they employ is included in the Appendix (p. 267).

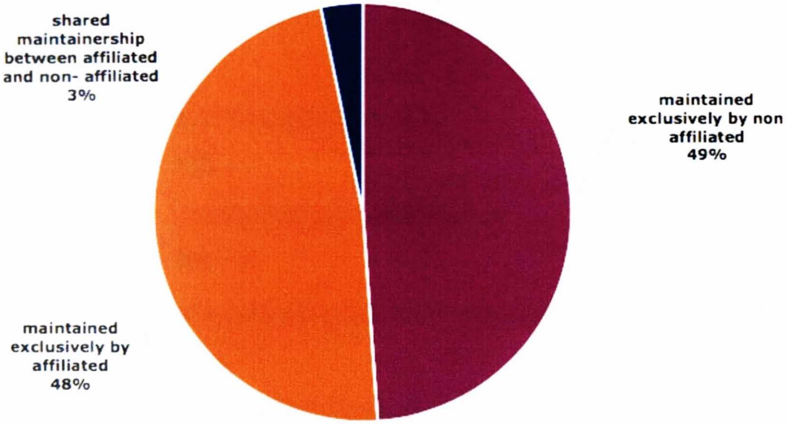


Figure 6-14 : Distribution of modules maintained by affiliated, non-affiliated, and affiliated and non-affiliated KDE maintainers

Source: Online search, N=191 (modules)

Examination of the patterns of maintainership across the different areas of development, depicted in Figure 6-15 reveals that the majority of *Core/Platform* and *Main Desktop* modules are maintained by affiliated developers. The test statistic reveals that in cases of exclusive maintainership, these differences are statistically significant.[76] The chi-square of the crosstabulation in Table 6-16 has a value of χ^2 (df=3, N=176)= 26.656, p<0.0001. The counts included in Table 6-16 allow us to calculate the odds ratios indicating the likelihood that the maintainer of a module belonging to these critical areas of development is a paid developer. The ratios indicate that a *Core/Platform* module is 2.35 times more likely to be maintained by a paid developer than a volunteer. At the same time, a *Main Desktop* module is 1.53 times more likely to maintained by an affiliated developer than a volunteer.

It should be noted that the analysis did not take into account, QT, KDE's graphical toolkit, that was developed by Trolltech ASA, since the package is not included in KDE's main release. If it had been included then the proportion of corporate presence in *Core/Platform* modules would have been much higher.

Figure 6-15 also indicates that user-productivity applications (*Apps*) and modules associated with *Development Tools* are maintained predominantly by volunteers. This result stands in contrast with the findings for the KDE e.V. members which indicated that paid developers are more active in this aspect of development. The odds ratios (obtained from Table 6-16 calculated for the two groups of developers) inform us that an *Application* is 4.21 times more likely to be maintained by

[76] For the justification for focusing on cases of exclusive maintainership see p.15, footnote 16.

a volunteer than by an affiliated developer. Lastly, a module associated with *Development Tools* is 2.44 times more likely to be maintained by a volunteer than by an affiliated developer.

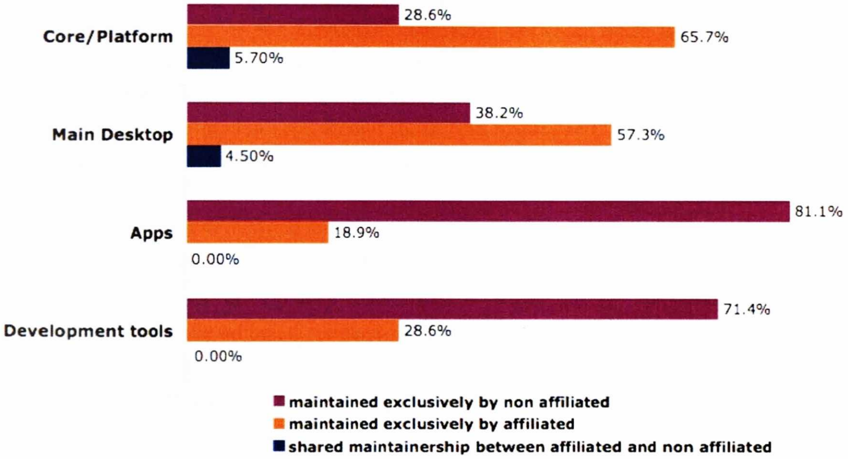


Figure 6-15: Analysis of modules maintained by affiliated and non-affiliated KDE developers in the principal areas of development

Source: Online search, N=191 (modules)

Maintainership Area by Affiliation/ CROSSTABULATION

		Area of Module Maintained				
		Platform	**Main Desktop**	**Secondary Desktop-Apps**	**Developer tools**	**Total**
Maintained exclusively by non- affiliated	Count	10	34	30	15	89
	Expected Count	16.7	43	18.7	10.6	89
	% within the Group	11.2%	38.2%	33.7%	16.9%	100.0%
	% within the Area	30.3%	40.0%	81.1%	71.4%	50.6%
	Adjusted Residual	-2.6	-2.7	4.2	2	
Maintained exclusively by affiliated	Count	23	51	7	6	87
	Expected Count	16.3	42	18.3	10.4	87
	% within the Group	26.4%	58.6%	8.0%	6.9%	100.0%
	% within the Area	69.7%	60.0%	18.9%	28.6%	49.4%
	Adjusted Residual	2.6	2.7	-4.2	-2	

Table 6-16: Crosstabulation of cases of exclusive maintainership*area

Note: these are modules maintained exclusively by non-affiliated and modules maintained exclusively by affilitated and the areas to which the modules belong

Source: Online search, N=191 (modules)

Consistent with the findings from the KDE e.V. network, and in line with the results of the analysis of the GNOME maintainer network, there is a significant association between affiliation and number of modules maintained. For the purpose of the analysis developers were divided into three groups. The first group included those who maintained one module, the second those who maintained two modules and the third those who maintained more than two modules. As indicated in Figure 6-16 whereas volunteers frequently maintain only one module compared to non-affiliated developers, affiliated developers more commonly maintain more than two modules. In fact, the relationship between affiliation and the specified grouping of contributors according to number of modules maintained is significant (χ^2 =(2, N=111)=7.168, $p<0.05$). However, the adjusted residuals included in Table 6-17 show that the most important association is between developers maintaining only one module. The odds ratio indicates that a paid developer is 62% less likely to maintain only one module than a volunteer.

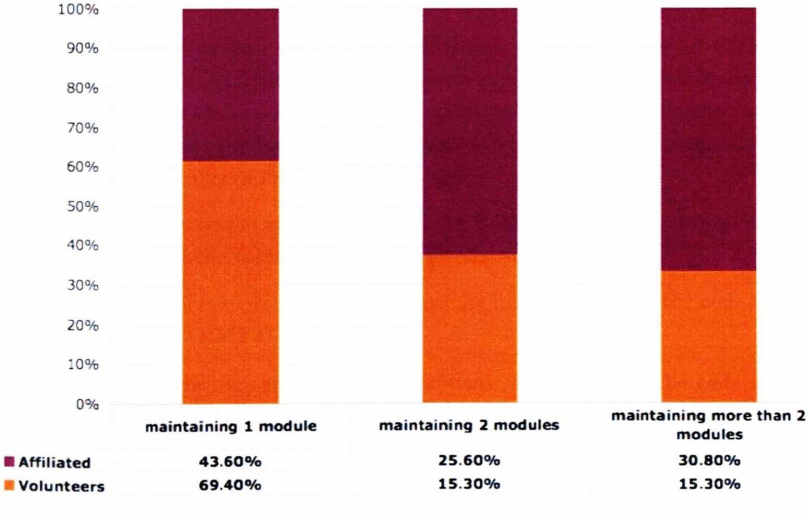


Figure 6-16: Distribution of number of modules maintained by non-affiliated and affiliated KDE developers

Source: online search, N=111 (maintainers)

No of modules maintained by affiliation/ CROSSTABULATION

Group Category according to no of modules maintained

		Maintaining 1 module	Maintaining 2 modules	Maintaining more than 2 modules	Total
Non-affiliated	Count	50	11	11	72
	Expected Count	43.5	13.6	14.9	72
	% within the Group	69.4%	15.3%	15.3%	100.0%
	% within the Group Category	74.6%	52.4%	47.8%	64.9%
	Adjusted Residual	2.7	-1.3	-1.9	
Affiliated	Count	17	10	12	39
	Expected Count	23.5	7.4	8.1	39
	% within the Group	43.6%	25.6%	30.8%	100.0%
	% within the Group Category	25.4%	47.6%	52.2%	35.1%
	Adjusted Residual	-2.7	1.3	1.9	

Table 6-17: Crosstabulation of affiliation with number of modules maintained

Source: Online search, N=111 (maintainers)

In summary, despite the fact that KDE has a smaller number of affiliated maintainers than GNOME, the findings for the two projects are consistent in that the most critical aspects of the code base, *Core/Platform* and *Main Desktop* modules, are in both cases maintained predominantly by affiliated developers who also consistently maintain a larger number of modules than volunteers.

Based on the results of the analysis for the KDE project, the following can be stated for the hypotheses formulated at the beginning of this chapter

- Paid developers are more likely to contribute to critical parts of the code base. **DISCONFIRMED**
- Paid developers are more likely to maintain critical parts of the code base. **CONFIRMED**
- Volunteer contributors are more likely to participate in aspects of the project that are more geared towards the end-user. **DISCONFIRMED**
- Programmers and peripheral contributors are not likely to participate equally in major community events. **CONFIRMED**

The next section recapitulates the similarities and differences between the two projects and discusses the generalizability of the findings.

6.5 Conclusion

This section relates the findings to the hypotheses outlined at the beginning of the chapter and draws out their implications for this research.

These hypotheses were that:

1. Paid developers are more likely to contribute to critical parts of the code base.
2. Paid developers are more likely to maintain critical parts of the code base.
3. Volunteer contributors are more likely to participate in aspects of the project that are more geared towards the end-user.
4. Programmers and peripheral contributors are unlikely to participate equally in major community events.

The findings complement the insights from the qualitative examination of commercialization and peripheral participation by highlighting the importance of instititutional support for the division of labour in F/OS communities as well as for creating enhanced opportunities for learning and community involvement. This section explores some of the limitations of the methodology in the light of the insights gained from the data analysis. First, it is necessary to recapitulate the key findings of the analysis.

The examination of the GNOME community reveals that volunteer and paid contributors modes of involvement are distinctive with respect to primary and long-term patterns of contribution, community events attendance and the number and types of projects they maintain. In addition to confirming the presence, as indicated by other studies (Koch and Schneider, 2002; Lee and Cole, 2003; Mockus, et al., 2002), of a fairly small group of core developers, that is, programmers that are regularly involved in key aspects of the development, the analysis shows that these individuals are predominantly hired developers employed to work either specifically on GNOME, on GNOME and other F/OS projects. Examination of the GNOME Foundation indicated that the most pronounced differences are between volunteers and developers hired to work on GNOME (either on an exclusive basis or in combination with working on other F/OS projects). This suggests that the key differentiating variable for the groups examined is 'being employed to work on GNOME'. Examination of the GNOME maintainer network confirmed the leading role of paid developers in maintaining infrastructural aspects of the project.

When combined, the results for the two GNOME networks indicate that hired developers: a) contribute to the most critical, infrastructural aspects of platform and desktop technologies (*Core/Platform* and *Main Desktop* Modules); b) maintain most modules in the area of *Core/Platform;* and c) maintain a larger number modules than volunteers. Analysis of the data from the GNOME Foundation indicates that programmers hired to work on GNOME maintain more modules than volunteers and programmers employed to work on F/OS but not on GNOME. Finally, this part of the study indicated that paid developers employed to work on GNOME attend considerably more major community events than volunteers and other paid hackers. In contrast, patterns of involvement, attendance and maintainership between volunteers and developers who are employed to work on F/OS but not on GNOME are not dissimilar. At the same time, the data analysis indicates that volunteers are consistently involved in, and maintain more end-user oriented and peripheral aspects of development. These findings confirm all four of the initial hypotheses.

The case of KDE is rather different at the organizational level, but consistent with GNOME with regard to patterns of maintainership. Volunteer KDE e.V. members reported to have been more active than paid developers in the areas of *Core/Platform* and *Main Desktop*. However, neither the pattern of primary (main) nor long-term contributions was statistically significant for any of the examined groups. Examination of the KDE e.V. network highlights important differences between affiliated and non-affiliated developers with regard to: a) attendance at community events; and b) number of modules maintained. Specifically, there was a significant association between a) a high attendance record and being employed to work on KDE and KDE and F/OS, and b) between affiliated developers in general and maintaining more than one module. The study of the KDE maintainer network confirms the importance of the association between affiliation and number of modules maintained and indicates that whereas *Core/Platform* and *Main Desktop* modules are more likely to be maintained by affiliated developers, *User-oriented Applications* and *Development Tools* are more likely to be maintained by non-affiliated contributors. These findings confirm hypotheses 2 and 4.

What are the implications of these findings for the study of commercialization, peripheral participation and the study of F/OS communities as constellations of practice? The findings lend support to the following observations:

The distinctiveness of volunteer/peripheral and affiliated/core communities:

Three out of the four networks analyzed (the GNOME Foundation, the GNOME maintainer network and the KDE maintainer network), indicate that there is a strong association between paid developers and core hackers and between peripheral developers and volunteers. Another finding that corroborates this is the very small number in both projects of modules that are maintained via cooperation between volunteers and hired developers ('shared maintainership'). This finding supports the view that F/OS communities are not simply locales for learning and experimentation but are also spaces shaped by the demands of continuous production (see Chapter 5).

The importance of corporate contributions in project development:

The significance of paid developers' contributions and their role in maintaining crucial aspects of the code base, which indicates that these individuals have been involved for some time in project development, is consistent with the view of companies as significant *internal* actors in project development (see Chapter 4).

The importance of institutional support for learning and sustained contribution:

Although it is reasonable to assume that the involvement of paid contributors in critical parts of the code base is related to their employers' priorities, strategies and business models, there may be an additional explanation for their increased involvement in this central area of development. Substantial and continuous contributions in infrastructural modules, such as desktop and platform libraries, require a high degree of knowledge and technical competence. The ability to work full-time on a project allows paid developers to develop their technical skills and their understanding of the code base to a greater extent than volunteers who usually contribute in their free time. The limitations of the study methodology do not allow us to quantify commits, to test, for example, whether the submissions made by KDE e.V. volunteers in these areas are incremental or substantive.

The importance of institutional support for community involvement:

The fact that many volunteers cannot afford to attend community events has been acknowledged by the developer communities. Both KDE e.V. and the GNOME Foundation distribute funds annually to cover travel and accommodation expenses for many contributors who would otherwise be unable to attend such events. Besides socializing, conferences offer significant opportunities for skills development and often function as hotbeds for new ideas and projects. Although proximity is often considered to be a central characteristic of CoP, the importance of offline events has been ignored to a great extent in the F/OS literature. The views of a central GNOME contributor on GUADEC suggest that such events contribute to the impression of the stratified character of F/OS communities:

> Well, what I am kind of referring to is in meetings like GUADEC where you will have a group like, a big bunch of friends basically who get together every year and, you know, most of them are quite active developers and stuff like, but there are quite a lot who kind of used to be active developers and are not so active any more, so in that sense, that's almost like a big bunch of friends where you would think about the actual development usually, and then you'll have much more newer people involved, people who, you know, never have the opportunity to go to GUADEC and stuff like that. So it's almost like the community splits in two: what you see in GUADEC and the actual development community?
>
> (Martin, 11/07/04 p. 23)

The picture presented here would be incomplete if it did not take into account the following points that emerged through the process of analysis:

Paid developers often maintain modules they were not employed to directly contribute to.

When presented with some of the initial results of the study, the three developers who assisted in the validation of the online data pointed out that employed developers, especially community-integrated ones (see Chapter 4), often maintain modules that they are not paid to contribute to. When asked why this happens, a GNOME contributor said:

> It's mostly that people work on a module in their spare time, or are sponsored by their employer to work on a module, but then they get assigned to other work by their employer and continue to work on the module either out of a sense of responsibility or because they enjoy it. That might sound like "evil employer screws up project by re-assigning developers", but in reality the developers themselves might be looking for some new challenge to work on and are tired of their work on GNOME e.g. right now, I'm grabbing an hour so I can run through some un-reviewed GNOME patches that cannot be committed without my review since no-one has stepped up to maintain the modules in question since was re-assigned by RedHat to different work.
>
> (Martin, email sent on 4 May 2006)

A KDE developer added that:

> It's sometimes hard to separate between paid work and volunteer work for KDE. It both goes together and the boundaries are not always clear. I'm not unhappy with that, though.
>
> (CS, KDE e.V. survey email sent on 7 July 2005)

There are other layers of institutional support that can elucidate further the relation between employment and contribution in critical aspects of development.

The process of identifying maintainers and their affiliation revealed that many were in fact associated with high level education and research institutions, either as students (many of whom had studied at postgraduate level) or employees. This study did not allow for these types of affiliation to be taken into account. A more comprehensive investigation should examine different types of institutional support. This would enhance our understanding of employment relationship in F/OS communities by highlighting, for example, the professional development of contributors who are on a 'critical development path' (see Chapter 4). Thiago Maceira one of the volunteers who had contributed continuously to KDE core modules, was employed in 2005 by Trolltech ASA. Since January 2006 the company, after opening a new office in Berlin, had hired many more KDE developers. It would be interesting to know whether these individuals were among the key contributors to the project.

The generalizability of the GNOME Foundation and KDE e.V. data is framed by their membership rules and evolution (see Chapter 3), but at the same time, the findings highlight the way these institutions are perceived by community members.

It appears that the perceptions underlying participation in the the GNOME Foundation and KDE e.V. are different. The GNOME Foundation, which has been operational for longer than KDE, is frequently regarded by many developers as an extension of the community. An interviewee, who is also on the Foundation's, membership committee remarked that:

> [....and I think this is what is going on. And I do not think it is a good thing, but people tend to think that the Foundation is community, so they just want to be in the Foundation, but they are not always interested in what's going on in the Foundation, so...
>
> (Gerri, 18/10/04, p. 13)

On the other hand membership in KDE e.V. is often framed in more technical, administrative terms.

> Yes. My KDE e.V. membership came quite late, but there's a good reason for that. I felt completely comfortable with the openness the other developers showed me, in IRC, mailing lists, etc., so I didn't realise there was something to the e.V. All I knew was that they had the trademark for "KDE" and decided the location of the next aKademy. On the other hand, that means the e.V. doesn't communicate very well its objectives.
>
> (TM, KDE e.V. survey email sent on 4 July 2005)

This, in combination with KDE e.V.'s stricter membership rules could explain some of the differences in the patterns of contribution of KDE e.V. and Foundation members. A more dynamic examination of the evolution of membership in these institutions and its comparison with a more widely drawn sample of contributors would help establish whether, in time, these two bodies evolve to represent more accurately represent the community populations.

This chapter has highlighted F/OS communities as stratified constellations of practice, where employment relations are strongly associated with divisions of labour and patterns of community engagement. The findings challenge oversocialized accounts of participation and community organization and further our understanding of corporate involvement in F/OS communities (Chapters 4 and 5). The view that emerges is one in which F/OS communities are far from horizontal, user-driven innovation networks as indicated by von Hippel (2002) and others. They seem rather to be complex spaces shaped by the demands of production which incorporate different agendas and demands. Chapter 7 weaves together the empirical findings from this study and discusses their wider theoretical significance.

7 Technologies of communities and the relational meshwork: Principal findings and theoretical implications of the research

7.1 *Chapter overview*

The aim of this chapter is to weave together the empirical findings of the study and draw out their theoretical implications. The findings highlight F/OS collaborative communities as constellations of practice, permeated by different priorities, and signification and institutional support frameworks that create a complex topology of participation and collaboration. This topology is grounded in the demands of the gift and exchange economies and the priorities of the various groups in the community. At the same time it is shaped by discourses and practices constitutive of 'technologies of communities'. The term technology of communities signifies specific techniques, tools and processes that are central to establishing communities as autonomous spaces and as subjects and objects of governance and management. The intersection of technologies of communities with the various demands and priorities negotiated within the context of F/OS projects, creates different opportunities for agency and co-option. In this chapter it is argued that these opportunities may be indicative of a novel configuration of sociality and economic production supported by ICTs.

The chapter is structured as follows. Section 7.2 provides the first-level analysis of the empirical findings: it reviews the key findings of the research and discusses their implications for the empirical investigation of mature F/OS community-led projects within the context of the F/OS literature. Section 7.3, which provides the second-level analysis of the findings, builds on the themes and insights from the previous section to highlight the wider theoretical implications of the research, mainly the extent to which F/OS communities are indicative of a new model of power. Section 7.4 explores some alternative frameworks for the interpretation of the findings, such as Systems of Innovation (SI) and New Institutional Economics (NIE). Section 7.5 summarizes the key empirical findings and the theoretical contribution of the study.

7.2 *Commercialization and peripheral participation in mature community-led F/OS projects: Recapitulation and discussion of empirical findings*

7.2.1 *Recapitulation of research findings*

In this section the main qualitative and quantitative findings presented in Chapters 4, 5 and 6, are related to the original empirical questions and their implications for the investigation of F/OS development in relation to the dominant view of F/OS are discussed. The research was guided by the following questions:

- What are the dynamics that underlie F/OS software projects?
- How are power relations between various actors constituted and maintained at different levels of interaction?

The results of this study suggest that commercialization and peripheral participation, which in the context of this study refers to the integration of new programmers and the role of non-programmers, are critical for the organization of power relations within F/OS communities. This mode of operationalization follows Foucault's suggestion about how to analyse power relations (see Chapter 3) and is underlain by the examination of blind spots to participation where it is used to contest the established modes of rationalization of F/OS development. In investigating the dynamics of commercialization, the following questions were formulated:

First, what is the impact of commercialization in project dynamics?

In examining how the gift and the exchange economies intertwine the research:

- Examined the role of paid developers and the way the boundaries between companies and the communities are managed and negotiated.
- Highlighted the new interfaces, such as bounty hunt contests, that are being created between corporate players and the community.

Second, with regard to centrality and peripherality the study enquired into how centrality and peripherality, two central concepts of the COP perspective, are experienced and constructed and examined whether the discourses and practices associated with each of these two spheres of activity are similar or distinctive.

These questions were operationalized by:

- Investigating the role of learning in structuring relations between newbies and senior developers.
- Highlighting the organization of non-programming teams, particularly translators and documenters, and investigating their patterns of cooperation with coders.

Examination of the dynamics of commercialization indicates that:

The gift and exchange economies are interdependent

Analysis of the interview data confirms previous research indicating that communities have become more responsive to accommodating commercial needs. The normalization of the development process and the establishment of specific organizational bodies entrusted to liaise between communities and commercial actors are developments that have been associated with the accommodation of commercial cycles of development and legal requirements. In Chapter 4 it was argued that this view is consistent with the idea of corporate players as actors external to the community. However, in examining the role of paid contributors in community initiated and led F/OS projects it was revealed that relations between the two economic spheres are far more complex than this perspective might suggest.

The analysis indicates that employed developers with close community ties, and especially those that previously worked as volunteers in a project, embody a network of connections and extensive know-how about community processes, which facilitate acceptance of their work and collaboration with other community members. At the same time, these individuals are sensitive about balancing corporate-community interests; this is especially so for developers working in companies with rather weak ties to the F/OS world (see Chapter 4).

The quantitative findings in Chapter 6 confirmed the importance of the role of paid developers. The analysis indicates that these individuals are involved in and maintain crucial infrastructural aspects of development, and participate more regularly than volunteers in major community events. It was argued that, compared to volunteers, paid developers are in a better position to cultivate the knowledge and technical competence required to enable a substantial contribution to critical parts of the code base, since they can work on projects full time. However, the link between employment and involvement in core aspects of development needs further examination since companies frequently hire volunteers who are already on a 'critical path'[77] in the project's development.

These findings suggest that the involvement of corporate actors in the socio-technical fabric of the community through the employment of developers with close community ties, is mitigated by the restrictions placed on these individuals in terms of grounding work relations in social relations, indicated by the fact that many paid developers maintain modules that they were not directly employed to contribute to. In addition, these findings tend to confirm the argument that institutional support appears to play an important role in project development and community life, since it allows contributors to develop their expertise and the network of connections required to mobilize the help of other contributors. Institutional support might be particularly important for integrating infrequent contributions enabled by the modular and granular character of F/OS projects.

The boundaries between the gift and exchange economies are permeable and their respective needs and requirements are constantly renegotiated in the context of projects

The interdependence of the two economies is expressed at an individual level through the blurring of boundaries between tasks performed within the context of employment and community life, and at a higher level with regard to how corporate and community requirements are taken account of and balanced.

[77] Being on a 'critical path' means having already demonstrated technical competence and involvement in critical aspects of developments. For more on this see Chapter 4, section 4.3.2.

The research shows that the main dynamics underlying this process are:

- The need to maintain stability and encourage experimentation. The use of projects in a production environment puts pressure on the community to deliver a stable code base. At the same time, however, it is essential that developers have the freedom to experiment, since this is one of the primary motives for joining F/OS projects.
- The need to cultivate and maintain a high level of technical excellence without raising the barriers to access to the point where they discourage new volunteers from joining projects. In Chapter 4 it was argued that it is to the benefit of companies as much as to communities to retain the volunteer aspect of projects and maintain the balance in favour of the gift economy.

These two dynamics form an integral part of the ***instrumental modes*** of development in F/OS, i.e the underlying dynamics of how power operates (see Chapter 2, section 2.3.2. and Chapter 3, section 3.4.1).

Discussion of the GNOME Bounty Hunt and the GStreamer/Fluendo cases highlighted several aspects of the relationship between market and social production. The case of GStreamer/Fluendo pointed to the difficulties involved in making the transition from a purely volunteer based project to one involving corporate players. The faster rate of development, spurred by the involvement of paid developers can alienate volunteers who are unable to dedicate the time needed to keep up to speed with the changes made to the code base. In addition, the hiring of a number of key Gstreamer programmers by Fluendo SL, blurred the distinctions between company and community.

The GNOME Bounty Hunt contest prompted a debate about the extent to which commercial requirements, the Novell Corporation's in this case, should be responded to, and whether monetary incentives could undermine the volunteer basis of projects. It also raised some interesting questions with regard to whether and how these types of initiatives bring in new contributors or whether they simply reallocate resources and reshuffle priorities within the community. The GNOME Bounty Hunt contest revealed that initiatives of this type raise interesting issues with regard to how bounties are defined and managed and their impact on community development˙

This aspect of the study indicates that whose requirements take priority in the development process is not always straightforward. The involvement of integrated paid programmers substantially shapes the development agenda, albeit in a less obvious way than do bounty contests. On the other hand, such contests reveal the complexity involved in defining and implementing a coherent and transparent development agenda and pose the question of how different voices and interests are represented within projects. They also reveal F/OS projects to be not spaces of abundance, but sites where community resources need to be managed. This issue is discussed in the next section, where the question of communities as objects and subjects of governance is examined in more detail.

The investigation of peripheral participation highlighted two different forms: legitimate peripheral participation and autonomous peripheral participation, which describes the activities designed to complement code development within the project. The research indicated that:

Legitimate peripheral participation is shaped by the demands of production.

The analysis indicates that joining mature F/OS projects is not simply a process of enculturation, but is shaped by the demands of constant production and underpinned by significant barriers to access. The research examined the issue of learning and integration from the perspectives of newbies - new developers and senior developers (Chapter 5, section 5.3). More specifically, the results suggest that although communities make significant efforts to lower the barriers to participation, newbies face three types of difficulties: 1) difficulties associated with familiarizing themselves with the tools of F/OS development; 2) conceptual difficulties related to understanding how things are set up, and how they fit and are put together; and 3) difficulties related to how they present and situate themselves in the development, that is, what tasks they choose and how they ask for support.

The analysis in Chapter 5 indicated that self-presentation is a crucial element in mobilizing help from senior developers. Established members of the community usually prefer individuals who are able to clearly identify the tasks they want to undertake and who can demonstrate an understanding of the issues and processes involved. These expectations are related to two key characteristics of new contributors that senior developers value: self-reliance and commitment. The high turn-over of contributors and the significant demands made on the time of experienced developers seem to be the main factors shaping these expectations. In brief, the findings in Chapter 5 indicate that:

Although one-off or sporadic contributions are appreciated, senior developers prefer to dedicate time to helping individuals who are productive, do not need a great deal of guidance, and who are willing to commit to the process in the long term. The practices and schemas employed by these groups are indicative of the conditions of reciprocity underlying participation and of achieving acceptance by the centre of the community.

In mature F/OS communities legitimate peripheral participation is expected to be a solitary activity that takes place in the background not the forefront of development.

Learning relations are principally relations of production: they are embedded in a framework of cooperation shaped by the demands of continuous production, as reflected in the F/OS definition of meritocracy, and they are articulated through the social organization of the division of labour.

These findings confirm Berguist and Ljungberg's (2001) conclusions about the conditions of gift exchange in F/OS communities and the connection with power relations at a social level (see Chapter 2). The key insight from this aspect of the research, however, relates to the important link between learning relations, work and power, an aspect of CoP that has been significantly under-theorized in the literature (Contu and Willmott, 2003).

Programming and non-programming teams are distinctive in their priorities, make-up and rhythms of participation

This study has developed the term autonomous peripherality to encapsulate the unique characteristics and organization of non-programming activities within F/OS projects. In Chapter 5 it was argued that autonomous peripherality is not simply a preparatory stage to joining the main programming community, but is a distinct sphere of activity. In this context, the organization of translation and documentation teams and their cooperation with programmers, were examined in Chapter 5, section 5.5.1. Analysis of the qualitative data indicates that:

- Contributions made by translators and documenters are seasonal in character: they usually intensify during freezes, periods preceding major releases where development stops, which allows these teams to catch up with the progress made by the coding teams and finalize their work.

- There is an inherent tension in accommodating the needs of translators and documenters which requires a non-active phase of development and a willingness on the part of coders to resist tinkering with code right up to the last minute prior to a release.

- Whereas programmers focus on technical excellence and experimentation, the discourse of non-programming teams is underpinned by the value of access and ease of use.

- There is a tension between the need to invite contributions, which is associated with the effort to lower the barriers to participation, and the need to ensure high quality.

- Although the value of non-coding contributors is increasingly being recognized, it is widely acknowledged that peripheral contributors are less likely to achieve a high status within the community than programmers.

- Lastly, the quantitative analysis in Chapter 5 indicates that there is a strong association between peripheral contributors and volunteers in terms of their primary area of activity and that in this regard these teams of contributors are quite distinct from the core programming community, which is strongly associated with paid contributors, especially at the level of maintainers.

In short, the analysis indicates that rather than forming a homogenous community of developers, mature F/OS communities are comprised of constellations of differing and hierarchically organized communities of practice, which come together through the project's overarching goals, but which have distinctive priorities, rhythms of development and patterns of collaboration.

The need to maintain a coherent community base that is both welcoming to non-programming contributors and newbies and does not alienate the technical sector of the community constitutes another *instrumental mode* of power in F/OS.

The above provides a more nuanced view of F/OS communities than that based on the dichotomy between developers and users. Although the involvement and importance of peripheral teams will necessarily depend upon the maturity and the intended users of F/OS projects, there is a need to develop more elaborate conceptual models of centrality and peripherality for communities focusing on the production of complex goods.

Figure 7-1 summarizes the main insights afforded by the first level analysis of the research findings.

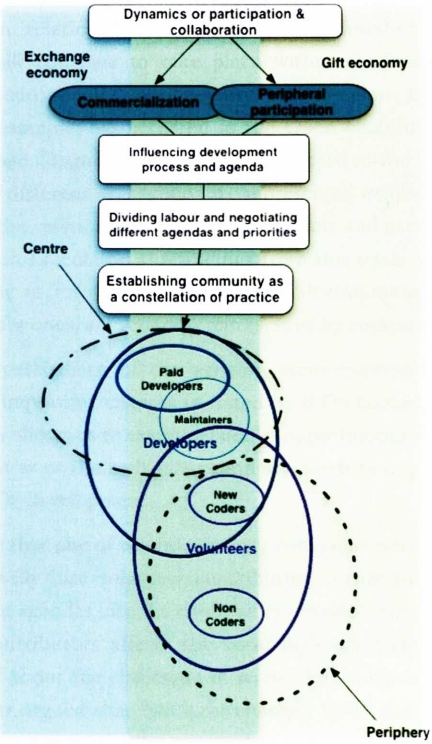

Dynamics or participation & collaboration
Exchange economy
Gift economy
Commercialization
Peripheral participation
Influencing development process and agenda
Dividing labour and negotiating different agendas and priorities
Centre
Establishing community as a constellation of practice
Paid Developers
Maintainers
Developers
New Coders
Volunteers
Non Coders
Periphery

Figure 7-1:Insights from first level analysis of commercialization and peripherality

Commercialization, peripherality and opportunities for agency and co-option.

The first level analysis suggests that centrality and peripherality are associated with the division of labour, which, in turn, is associated with employment relations and the institutional support frameworks. Is there, however, a deeper connection between commercialization and peripheral participation, especially with regard to the long-term dynamics of F/OS development? In Chapter 5 it was suggested that the development of autonomous peripherality is related to the evolution of F/OS communities.[78] Another important finding is that, although F/OS development belongs to the tradition of unrenumerated collaboration dating back to the early days of the Internet, it is also embedded in the dynamics of the commercial world and the associated broadening of the F/OS software user and contributor base.

In this context it is suggested that the evolution of F/OS development is following a similar trajectory to Internet evolution and the development of other historical large-scale infrastructures (Braudel, 1982). This trajectory is characterized by a shift from the largely elitist bases of such systems to utilization by a wider population through the persistence of existing socio-economic structures. An interesting question in relation to F/OS evolution is whether F/OS software's supportive activities will continue to take place within the context of technical projects or whether autonomous peripherality will develop its own, separate infrastructures as, for example, has occurred in the Open Usability Project and the Launch Pad initiative (see Chapter 5). Another issue related to the wider adoption of F/OS concerns the way different user groups are represented within projects. Despite the dominant view of the relationship between developers and users as being one of direct communication and feedback, the findings from this study suggest that there are larger forces at play in formulating the projects' development agenda and that users, especially corporate ones, are generally represented by commercial actors.

Alongside the persistence of the existing socio-economic structures, it is important to note the important changes initiated by F/OS communities. Foucault's relational view of power allows us to examine parallel opportunities for agency and co-option, and to form a view of the ambiguities and asymmetries implicit in the power relations underlying F/OS development.

We have shown that one of the reasons why companies with interests in F/OS projects prefer to formally hire volunteer contributors is that this allows them to integrate their corporate agendas into the community development process. However, the hiring of these contributors affects the company's organizational culture and processes. When asked about the challenges in reconciling company and community needs a KDE developer argued that "we have changed SUSE more than SUSE has

[78] See Chapter 5, section 5.4.

changed us" (Dean, p.6). This feeling of empowerment is usually reflected in the role(s) that paid developers assume within the context of their organizations and the tactics they employ to keep their professional and community involvement aligned (see Chapter 4). As indicated in Chapter 4, in large organizations with less prominent F/OS agendas, these individuals often assume the roles of educators of and/or mediators between the community, and the corporate teams that are unfamiliar with F/OS processes.

At the same time, the examination of the network of companies supporting members of the KDE e.V. and the GNOME Foundation and the two communities' maintainer networks (Chapter 6) revealed the business environment that is developing as a result of the commercial appropriation of these projects. Although this environment includes some large players, it is largely characterized by a substantial number of small and medium, and even micro-enterprises (built around one or two developers), and public institutions. Although further research and a more systematic examination is needed to full investigate the relations developing between these actors and their ties with F/OS, the current findings provide evidence of the work and entrepreneurship opportunities F/OS offers to individuals situated between the gift and exchange economies.

The increasing commercialization of F/OS software raises some interesting questions not only with regard to the future of F/OS communities, but also in relation to the impact of F/OS on the way companies conduct business. Consider, for example, the relationship between Trolltech ASA and KDE. In developing QT, a graphical toolkit available within a dual licensing model, Trolltech ASA largely depends on the KDE community for feedback. In a presentation given by Eirik-Chambe-Eng, one of the founders of Trolltech ASA, at the KDE academy at Ludwigburg in 2004, the relationship developing between the company and the community was described as synergistic. Eirik-Chambe-Eng highlighted KDE's role as representing a watershed for Trolltech ASA. He pointed to Trolltech ASA's policy of hiring KDE developers; KDE's role as a test-bed for Qt releases and a showcase for Qt. Although in this case, a special agreement was signed between the company and the community to ensure that KDE will always have access to updated versions of the toolkit, when regulating dependence at the development level, relations between community and corporate actors are often not formalized.

The examples, including that of Trolltech ASA, that have been cited in this analysis, highlight some interesting issues, such as what constitutes a mutually beneficial relationship between companies and communities and what are the requirements for an ethical F/OS software business. These questions have been debated by F/OS practitioners for a number of years (see for example: http://www.crynwr.com/cgi-bin/ezmlm-cgi?iis:8050:200209#b, last accessed 23/09/06).

How can we position the contribution of this study in the wider context of F/OS studies? As indicated in the Introduction (Chapter 1), many F/OS studies emphasize individual motivations (signalling competence, user-led innovation, etc.) and the desire to participate in a gift economy (giving unconditionally in order to benefit from the collective effort). This line of research is premised on the view that F/OS communities are composed of atomized actors whose behaviour might be influenced by the aggregate behaviour of others, or by the projects' governance structures. This strand of the F/OS literature translates the relation between the gift and exchange economy into one extrinsic and intrinsic motivations, monetary and non-monetary rewards, and their balance and interconnections (Bonaccorsi and Rossi, 2004; Franck and Jungwirth, 2003; Lakhani and Worf, 2005; Roberts, et al., 2006b) and is influenced by economists such as Bruno Frey (Frey and Jegen, 2001), who argue that monetary incentives may undermine or, under certain conditions strengthen intrinsic motivations ('crowding-out' theory).

A second strand of the literature is characterized by an oversocialized view of F/OS development and a consensual view of communities. Studies of this type emphasize the role of ideology and the values of the hacker culture in sustaining development, and usually favour accounts that ground the dynamics of cooperation and participation in social relations (Crowston and Howison, 2005; Elliot and Scacchi, 2003a; Hemetsberger, 2004; Kollock, 1999). It is interesting to note that these two strands of the literature complement each other in that they reproduce a strict separation between the two economies at the level of individual motivations.

The present study has addressed the deficiencies in these two strands of research by:

- offering an alternative account that counterbalances oversocialized explanations of participation, collaboration and learning that highlight the dependencies that form between different groups of contributors at different levels of interaction;
- extending the notion of peripherality - an idea developed within the context of CoP to take account of the complex division of labour in online, collaborative communities focusing on the production of complex goods;
- providing a basis for understanding how the gift economy is embedded in the exchange economy, which goes beyond individual motives.

Although the conceptual framework for this investigation was guided by Foucault's notion of relational power and his methodology for studying power relations, the analysis of the findings borrows from insights from a strand of the Economic Sociology literature and, specifically, from the body of work concerned with the notion of embeddedness originally developed by the American sociologist, Mark Granovetter (1985) to describe the significance of social relations in economic activities (see Chapter 2).

7.3 Technologies of communities and the relational meshwork: wider theoretical implications of the research

This section draws out the main theoretical implications of this research. To achieve this, it builds on the insights provided by the first-level analysis of the empirical findings to answer the overarching theoretical question of whether the F/OS model of development is indicative of a new form of power relation supported by ICTs. Based on the foregoing analysis the answer to this is a qualified yes. Although F/OS communities are far from constituting a complete knowledge/power system as defined by Foucault, they exemplify potentially important socio-economic transformations, which are underpinned by practices and discursive strategies evolving around the construction and appropriation of communities as separate socio-economic spaces with unique production capabilities.

In order to demonstrate how these transformations are supported, this section builds upon Nikola's Rose (1999) notion of 'technologies of communities' introduced in Chapter 4. Following the Foucauldian[79] definition, in this context the notion of technology refers to the techniques and processes that constitute specific programmes of action that give concrete form to specific rationalities. The *technologies of communities* discussed in this section include the following.

The programme of meritocracy. An explanation for the persistence of the meritocratic discourse as the prominent mode for rationalizing social order and hierarchy within F/OS projects despite the structurally biased character of the system indicated by the research findings.

The strategic and ritual enactment of the idea of community. Individual and team performances appealing to commonly held values and ideals, are essential for establishing and reifying the idea of a coherent community despite the often divergent agendas and priorities of different groups within the project, and for mobilizing volunteer resources, which, due to the proliferation of competing development agendas, can no longer be taken for granted.

Tools and techniques developed to address the emergence of a set of problems specific to community management and governance. This technology relates to the processes of objectification and more specifically to the tools required to visualize, survey and map community space and the techniques developed to manage labour. The knowledge generated through involvement in these processes is increasingly formalized as part of the effort to cultivate and manage volunteer communities.

[79] For the different meanings of technology specified by Foucault, see Chapter 2, section 2.3.2.

7.3.1 The program of meritocracy

Attention was drawn in the Introduction (Chapter 1) to the perception of F/OS communities as novel, self-organizing organizational forms sustained on the basis of a system of meritocracy that ensures that the status and power of an individual is accurately reflected in the importance assigned to his/her contributions. The specificity of the definition of meritocracy stands in contrast to the ambiguity of the political character of the movement that is populated by a range of different and, sometimes contrasting, ideologies, including libertarian, anarchic or socialist ideals. In simpler terms, although the F/OS movement means different things to different people, the meaning of meritocracy in F/OS communities is quite specific. This section revisits the meritocratic *programme* (a Foucauldian notion that refers to this particular rationalization of the way F/OS communities operate) in the light of the empirical findings from this research, and argues that it is constitutive of the 'technologies of communities'.

The programme of F/OS communities is based on the idea that meritocracy establishes a more natural social hierarchy than that found in the context of traditional bureaucratic organizations. In their (2004) book *The Meritocracy Myth,* Stephen McNamee and Robert Miller suggest that the ideal of meritocracy masks an ideology of inequality. The acceptance of the system is predicated on the *belief* that it is fair since everyone seemingly has an equal or adequate chance to succeed. However, as in the case of other enduring programmes of action, the programme of meritocracy is never fulfilled in F/OS for the following reasons.

There is potential for the system to be abused by opportunistic individuals. One interviewee complained, for instance, that another programmer had added some minor changes to a substantial piece of code, which was then presented in the CVS system as his own. In her PhD thesis titled 'Three Ethical Moments in Debian' Gabriella Coleman (2005) highlights some of the anxieties of developers in relation to the corruptibility of meritocracy and demands for greater transparency, accountability and accessibility on the part of those in high positions in the community hierarchy.

The definition of meritocracy as technocracy is challenged. The definition of meritocracy as technocracy is related to the intensively technical character of projects. However, as this study indicates, the success of F/OS seems to threaten communities as pure spaces of technical creativity and experimentation. Commercial and wider user demands place their own restrictions on the hacker's pursuit of excellence, undermining the meaning of meritocracy as technocracy. This is reflected in the broadening of the meritocracy definition indicated by, for example, the changing membership requirements for the GNOME Foundation.

Structural biases are introduced in the system through employment relations and other types of preferential access. As the results of this study indicate paid developers have a greater opportunity to attain higher positions within their communities because they are able to work on projects on a full-time basis.

Why then does meritocracy and, more specifically, the definition of meritocracy on the basis of technocracy, provide such an enduring explanation of the organization of F/OS projects? It is suggested here that this is due to several characteristics.

First, the programme's effectiveness in establishing and sustaining a detailed system of differentiation that underpins the highly individualistic hacker culture. This is expressed through multiple types of membership, associated with a significant number of informal and formal groups with overlapping boundaries. Large F/OS communities are rife with technical and administrative committees, work groups organized on the basis of different activities and initiatives and, in some cases, mailing lists and IRC channels to which access is subject to various vetting procedures. This landscape shapes different opportunities for inclusion and exclusion which means that there is always something to achieve, something to learn, and opportunities for individuals to distinguish themselves.

Second, there is a resonance with programmers' perceptions of the value of their work and the contributions made by other groups of employees in the workplace. Similar to what occurs among other work groups within organizations, programmers place a higher premium on their skills, which are often framed in terms of a craft (Himanen, 2001). In addition, they consider their role as critical in terms of accountability, as they frequently constitute an obvious target for blame within organizations when there are failures, or something does not function as it is supposed to do (Schaefer, 2006). Furthermore, as Tracy Kidder's (2000) account illustrates, developers often feel that their pursuit of excellence is constrained by organizational politics and the need to accommodate the demands of other groups such as clients and managers. At first glance, F/OS projects appear to provide an ideal space for developers to concentrate on their work unhindered by the politics of the workplace and the demands of other stakeholders. The persistence of technocracy is indicated by the shared perception that coding skills are definitely more valuable than other skills and this is connected perhaps to the desire to maintain F/OS projects as pure, technical spaces.

Third, programmes function as the basis for upholding the community's autonomy, which is essential for ensuring consensus and promoting recruitment. Upholding the values of meritocracy is essential for maintaining the equitable basis of participation, and forms an essential part of team face-work (see Chapter, section 2.4.2.). As this research has shown, community members accept the contributions of non-integrated hired developers as long as they are subject to the same rigorous process of peer-review that applies to volunteers. The case of community-integrated paid developers is more complicated. Although their affiliation is not secret, they are often treated as if their employment does not really matter. For example, one interviewee indicated that on assuming a high organizational position, such as taking a place on the GNOME Board of Directors, hired programmers are not expected to represent anyone but themselves. It is possible that the reluctance to dispute the loyalty and merit of integrated-paid developers stems from the need to protect the meritocratic basis of participation. It is interesting to note that in Coleman's (2005) study the issue of meritocracy was linked to recruitment and the integration of new members.

This section has analysed the programme of meritocracy as a particular type of rationality. Its persistence, despite its apparent weaknesses, was discussed alongside other inconsistencies as they emerged from the research findings. The meritocratic programme thus emerges as: "the generalization and interconnection of different techniques themselves designed in response to localized requirements" (Foucault,2000a: 231)

7.3.2 Enacting community: maintaining unity and mobilizing resources

The second constitutive element of technologies of communities is the way the idea of community is employed and performed to mobilize resources and maintain a unified basis of participation. Following the dual dimension of Goffman's notion of 'face-work' (see Chapter 2, section 2.4.2.) it is suggested that this strategy has both ritual and strategic aspects: personal and team performances are essential in sustaining the idea of a coherent community, while allowing actors to position themselves strategically in their effort to mobilize volunteer resources.

An argument that recurs in this study is that despite corporate involvement, volunteer participation continues to form the backbone of F/OS projects. As an increasing number of volunteer contributors seek to make a living from their involvement in projects, and small and large companies attempt to reap the benefits of F/OS development, volunteer participation, arguably the most coveted resource of F/OS projects, can no longer be taken for granted. Most of those who are familiar with the dynamics of F/OS development understand that sustained exploitation requires sustained involvement through the grounding of contributors in the network of relations that permeate the community, continued reassurances that the community is a separate, autonomous space, and an understanding of how to appeal to common values in order to mobilize community resources.

However, what is included in the idea of community is not always self-evident. Does the notion of community, for example, include the user community alongside the developer, coder and peripheral communities? Does it include the social as well as the technical? As this research indicates these communities overlap in terms of their overarching goals, but are at the same time quite distinctive in terms of their make up and priorities. In addition, access to different groupings and teams within projects is structured. Current discourses emphasize the importance of the developer community over the value of autonomous peripheral participant contributions. Enactment of the idea of a unified community across different sites, and in spite or because of its multiplicity, is therefore essential to establish and sustain it as an autonomous space.

The significance of the enactment of a unified community is reflected in the importance placed on the image of the community that is projected and the processes of control underlying different types of membership and forms of representation. Control over official representation of the community is strictly regulated in the context of projects. For instance, modification rights for the GNOME and KDE websites are more difficult to obtain than access to the CVS. The existence of gated mailing lists can be also be seen as an essential part of maintaining the integrity of 'front stage' performances, in creating a separate backstage space where certain issues can be discussed more openly. In addition to the evolution of certain expressions of formal membership, such as recognition of the value of non-coding contributions, some supporting higher-level activities are becoming increasingly professionalized. The GNOME board, for example, recently advertised for a GNOME Foundation director of business development.

The invocation of the idea of community, however, takes on an additional, strategic significance based on the way different groups are perceived and depicted and affect one another within the parameters set by different contexts of interaction, that is, within more public or more private spaces. One of the individuals primarily responsible for the GNOME Bounty Hunt, for instance, suggested that the contest, among other things, enabled users to get closer to developers. This suggestion contradicts the established view of a direct relationship between users and programmers and highlights how the different perceptions of users and their needs can be mobilized as part of a specific agenda.

The connection of this technology of community with the programme of meritocracy lies in the need to maintain the primacy of the developer community while at the same time not alienating the peripheral community. We have shown that this can in part be achieved through the provision of a framework of access and representation, which is based on the definition of meritocracy as technocracy, but which does not preclude non-programmers outright and provides preferential access for coders.

7.3.3 Managing F/OS: tools, tactics and the formalization of knowledge on how to govern and manage communities

This section investigates the third element of technologies of communities which concerns the tools and techniques developed to address a set of problems associated with governance and management. The previous section focused on technologies and the tactics of representation; this section focuses on the tactics of objectification, that is, what it takes for communities to be constituted as subjects and objects of governance. This is examined from a dual perspective: one that is internal to communities and their constitution as autonomous spaces and one that is concerned with how F/OS communities could be managed from the perspective of actors that want to appropriate the benefits of F/OS. This examination does not include specific governance or management models, but focuses on the lower-level tools and techniques that are required for their exercise. It is argued that this technology is associated with the emergence of a body of knowledge focusing on the constitution and management of communities, not only in F/OS, but also perhaps in politics, science and the workplace.

Although not always framed in political terms (Coleman, 2004), the balance between openness and control, coordination and decentralization, political power, legal representation and technical authority, is constantly debated and reflected upon in F/OS projects. However, the issues of governance and management and their models, principles, and mechanisms, are inextricably related to efforts to survey, map and visualize community space (Rose, 1999). F/OS generates an endless stream of data that include various types of documentation, mail and CVS digests (reports of code contributions made to the CVS repositories), personalized individual, module and bug statistics, RSS feeds from developer blogs, and a complicated network of websites. Given the fluid nature of F/OS communities, it is certainly true that in the absence of these data it would be impossible to identify what needs to be managed, governed or represented. The generation and use of these data are prerequisites for: visualizing and organizing the development process; surveying and mapping participant networks in all their different expressions, (social, technical etc.); and for measuring performance and authenticating participation.

The data provide various opportunities for action and intervention on both an individual and a collective level. The ability to verify contributions by referring to CVS statistics, for example, is essential for affirming the meritocratic basis of participation. Similarly, the facility to visualize the development process, progress made and potential delays, is a necessary aspect of mobilizing participation.

In addition, F/OS contributors near the top of the community hierarchy develop their own tactics for managing and organizing labour. A high-level KDE programmer, for example, described how the publication of the release schedule together with the incremental freezes and the accompanying announcements at each stage of the process, were utilized to get people into that "freeze feeling", to develop a sense of urgency towards a looming deadline (Roger, 22/08/04). In an address to the Massachussets Software Council entitled 'The Mechanics of Open Source: Growing and Harvesting your Open Source Project' Nat Friedman and Miguel de Icaza (2005), the founders of GNOME, described some of the basic requirements for a successful F/OS project. These included the need to maintain a core working group and the benefit of providing concrete tasks for people to become involved in. Friedman and Icaza utilized the insights they gleaned from development in the GNOME project to set up the MONO platform, another F/OS project controlled by Novell Inc., which provides a F/OS implementation of Microsoft's .Net architecture.

This migration of knowledge acquired in the context of F/OS development combined with the perceived benefits of F/OS for commercial purposes, has given rise to a body of knowledge related to the constitution and management of volunteer F/OS communities (Haruvy, et al., 2003; O'Mahony and West, 2005). The continuing popularity of studies concerned with developers' motivations is an enduring theme indicative of the effort to better understand, and thereby control, participation. As F/OS software is increasingly regarded as a potential engine for innovation and development, connecting and cultivating F/OS communities becomes a priority issue for transnational, national and regional institutions. The significance of the role of F/OS communities as new spaces of innovation and production echoes parallel developments regarding the perceived importance of communities in areas as diverse as science, politics and the workplace (Cross and Parker, 2004; Knorr-Cetina, 1999) and the emergence of a body of knowledge geared towards cultivating and managing them (Wenger, et al., 2002).

Figure 7-2 outlines the main elements of technologies of communities, the strategies, methods, tools and tactics that constitute them, and their function within the context of F/OS communities.

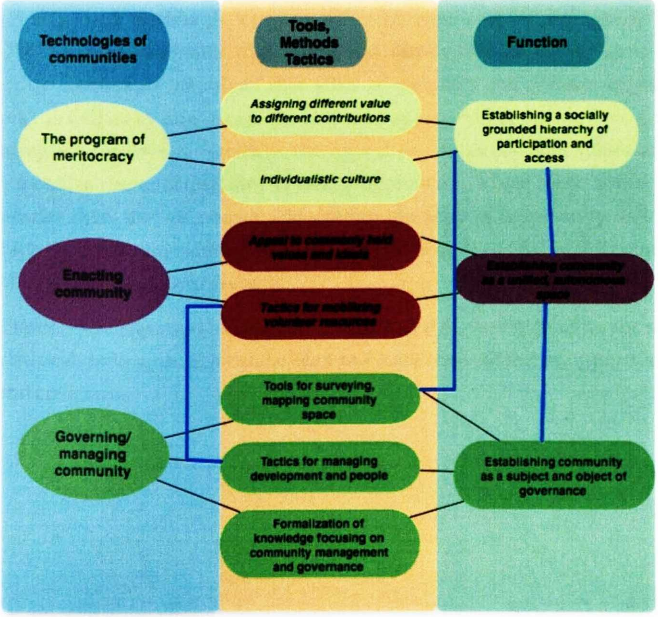

Technologies of communities
Tools, Methods Tactics
Function
Assigning different value to different contributions
Establishing a socially grounded hierarchy of participation and access
The program of meritocracy
Individualistic culture
Appeal to community held values and ideals
Enacting community
Establishing community as a unified, autonomous space
Tactics for mobilizing volunteer resources
Tools for surveying, mapping community space
Governing/ managing community
Tactics for managing development and people
Establishing community as a subject and object of governance
Formalization of knowledge focusing on community management and governance

Figure 7-2: Technologies of communities

A characteristic of technologies of communities is their inteconnected nature. For instance, as indicated, the tools for surveying and mapping community space are essential in the fulfilment of the programme of meritocracy and the tactics for managing development and people are closely linked to the tactics employed to mobilize volunteer resources. The related character of different elements of technologies of communities is also apparent from their different functions. For example, the establishment of community as a subject and object of governance presupposes its establishment as an autonomous space.

In this section the findings concerning commercialization and peripheral participation have been repositioned within the context of the wider theoretical questions that concern this study. How the programme of meritocracy and the techniques and strategies associated with differentiation, representation and the mobilization of resources establish a unified, widely accepted basis of participation that secures the idea of communities as coherent, autonomous spaces has been discussed. The tools for visualizing, mapping and surveying community space and management tactics have been shown to constitute communities as subjects and objects of governance and management. The interweaving of these technologies of communities with the structural and discursive forces associated with the demands of the gift and exchange economies, the dynamics of commercialization and peripheral participation, and other processes that were not examined in this study can be said to produce a relational meshwork, creating different opportunities for agency and co-option. For instance, the tools that enable the survey of community space can be employed by community members that want to pinpoint and address development bottlenecks as well as by companies interested in appropriating parts of the code base and scouting for employees. Similarly, the appeal to commonly held values and ideas can help mobilize the contributions of non-programmers, whose work is usually seen as less worthy than that of coders, and creates the idea of community as a unified space which becomes undermined when the priorities and agendas of different groups of contributors are taken into account.

Figure 7-3 brings together the two levels of analysis and highlights the presence of the relational meshwork that can be seen to create these different opportunities for agency and co-option.

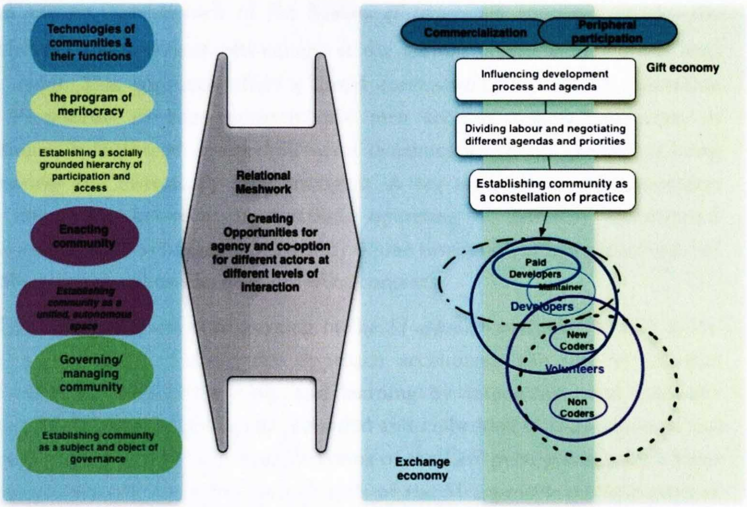

Technologies of communities & their functions
the program of meritocracy
Establishing a socially grounded hierarchy of participation and access
Enacting community
Establishing community as a unified, autonomous space
Governing/ managing community
Establishing community as a subject and object of governance
Relational Meshwork
Creating Opportunities for agency and co-option for different actors at different levels of interaction
Commercialization
Peripheral participation
Gift economy
Influencing development process and agenda
Dividing labour and negotiating different agendas and priorities
Establishing community as a constellation of practice
Paid Developers
Maintainer
Developers
New Coders
Volunteers
Non Coders
Exchange economy

Figure 7-3 The relational meshwork

7.4 *Alternative frameworks of interpretation and the baseline for comparing the F/OS model*

This section outlines approaches that might produce alternative interpretations of the research findings. These have been selected on the basis of their relevance to the emerging analytical themes and how they have informed other studies of the F/OS phenomenon. The first is the *Systems of Innovation approach.* Adopting an SI perspective implies examining F/OS communities as sources of technological and organizational innovation and positioning the findings in a context that focuses on the emergence and diffusion of knowledge between organizations and institutions (Edquist, 1997). The second is a *New Institutional economics* approach, which draws on the work of Oliver Williamson (Williamson, 1987; 2000; Williamson, 1979) on the nature and changing character of ecònomic organizations. Within the NIE perspective F/OS communities are examined in relation to markets and firms. The third perspective is drawn from studies in the field of geography and sociology. Building on the work of theorists, such as Nigel Thift (2005), Saskia Sassen (Sassen, 2004, 2005a, b, 2002) and Manuel Castells (Castells, 2000a, b) this approach interprets the insights afforded by this study in the context of new processes of commodification and changes affecting labour markets spurred by globalization and co-evolving technological and economic infrastructures.

This section evaluates the strengths and weaknesses of each approach against the insights afforded by the theoretical and conceptual framework outlined in Chapter 2 and applied in this study in order to reflect upon its validity.

In essence, one branch of the *Systems of Innovation* approach studies the process through which innovations emerge at the national, regional or sectoral level (Edquist, 1997). This approach offers a broad conceptualization of the innovation process. Proponents of this approach take into account a broad spectrum of organizational, institutional and technological dynamics which are treated as being interdependent and historically and contingent. A key argument is that innovation requires interaction between organizations operating in different institutional contexts and that success hinges on the ability of one institution to learn from another and be able to translate knowledge in different contexts.

The conceptualization of learning in the *SI approach* is similar to that in the CoP literature. Similar to CoP, this approach accentuates the role of informal learning, such as learning by doing and learning by interaction, and considers knowledge and learning to be socially grounded and embedded in organizational and institutional practices. Unlike most applications of the CoP perspective, which focus on inter-organizational dynamics, some strands of the SI approach are interested in the conditions that hinder or facilitate the scaling up of learning and capability building across institutions and organizations. The leading scholars in the field include Keith Pavitt, Christopher Freeman and Bengt-Åke Lundvall (Freeman, 1982; Lundvall, 2002; Pavitt, 1999).

Had this perspective informed the present study, developments spurred by commercialization and peripheral participation would have been examined as sources of sectoral, technological and organizational innovation and as engines for the mutual transformation of companies and F/OS communities. The importance of the role of paid developers as agents of learning and change in the organizations that employ them would have been emphasized, and developments such as the normalization of the release cycle and its organization in different 'freezes' would have been treated as instances of successful institutional and organizational adaptation. The *SI* approach also offers some interesting insights with regard to the long-term implications of the diffusion of F/OS infrastructures and the F/OS model of development. However, although the *SI* approach takes account of the institutional and organizational sources of conflict, it appears to lack a view of learning that takes account of its role in structuring power relations and the conceptual apparatus that is necessary to develop a more nuanced account of how different agendas and interests shape the dynamics of collaboration not just across, but also within, the same organizations and institutions.

In short, the weakness of this perspective in the context of the present study relates to the complex dynamics of collaboration and their associated tensions at the institutional level in a context such as F/OS where these dimensions are not fully formed. Foucault considered the examination of power relations at the level of institutions as a possible, but not particularly productive, avenue of research.

> One sees why the analysis of power relations within a society cannot be reduced to the the study of a series of institutions or even to the study of all those

> institutions that would merit the name "political". Power relations are rooted in the whole network of the social.
>
> (Foucault, 1982c:345)

In adopting a methodology that focuses on concrete strategies, conflicting and complementary programmes of action, conceptual schemas and tactics, rather than on formal institutional requirements and characteristics, this research offers a more nuanced account of how the emerging bazaar model is managed and how specific power relations are cultivated and reified, such as through the centre's co-option by commercial actors.

The same weakness in relation to the unified, formal view of institutions and organizations that underlies the *SI* approach applies to the potential, interpretative, value of the *NIE* tradition. *NIE* is mainly concerned with the study of economic institutions and examination of the institutional environment, constitutions, laws, property rights and governance structures that facilitate institutional alignment with economic exchanges, such as bureaucracies, firms and markets. One of the most prominent bodies of work in this tradition is *Transaction Cost Economics (TCE)*. The term transaction cost economics was introduced by Ronald Coase in his article 'The nature of the firm' (1937) to describe the costs of drafting and negotiating an agreement (ex ante costs), and the costs associated with the execution of agreements (handling and resolution of disputes and misalignments that may arise through their implementation, or ex post costs). Coase argued that firms, as novel forms or production organization, emerged because under certain conditions they provide more effective mechanisms than the price-induced mechanisms of the market for handling these costs. The *TCE* approach, therefore focuses on two interrelated aspects of economic activity: governance and effectiveness. Oliver Williamson (1987) specified the link between the two as follows:

> This is the world of governance. Since the efficacy of court ordering is problematic, contract execution falls heavily on the institutions of private ordering. This is the world with which transaction cost economics is concerned. The organizational imperative that emerges is this: Organize transactions so as to economize on bounded rationality while simultaneously safeguarding them against the hazards of opportunism.
>
> (Williamson, 1987:32).

The influence of the *TCE* approach on the study of F/OS has been twofold: on the one hand TCE has been used to establish a persistent research agenda, one that is concerned with the efficiency of the F/OS institutional framework in regulating and aligning contributors' motivations and the success of its coordinating mechanisms in overcoming conflict. This approach also informs debates on the relationship between private and public, and the compatibility of intellectual property rights, contract law, and established notions of ownership with safeguarding creativity in the digital age. It provides a point of departure for examining the specificity of F/OS, enabling researchers such as Yochai Benkler (2002) to adopt the *TCE* view of the firm as the baseline for comparing the F/OS model of development and its successes, with those of firms.

Had the NIE and TCE approaches been adopted in this study the focus of the analysis would have shifted to an examination of the divisions between coders and the peripheral community as an effective way of planning and allocating tasks, and might have interpreted the persistent barriers to access as mechanisms for minimizing the demands made on the time of senior developers. Such a view would supplement existing accounts of the effectiveness of the technical aspects of projects, for instance, their modularity, in planning and allocating resources. The involvement of paid developers along with other developments streamlining the development process, including the normalization of the release cycles, could be considered in relation to the governance structure of F/OS communities and their ability to scale up. At the same time, adopting these perspectives would mean assessing the role of integrated paid developers from the point of view of their role in aligning collective and commercial motivations and of minimizing potential conflicts arising from different development agendas. This would reposition the question of the relationship between the gift and exchange economies to one of compatibility between two different systems for regulating economic behaviour and associated forms of ownership and conflict resolution.

Another major weakness of NIE in addition to its unified view of organizations and institutions, is that it fails to take account of the wider context in which transactions are embedded. In our case, this includes the social dynamics of collaboration shaped by the gift economy and the interdependencies that develop between the gift and exchange economies. In short, although the NIE approach raises some important questions, it provides a rather instrumental and narrow framework for their examination.

The third alternative for the interpretation of the findings in this study draws on sociology and geography and builds on the work of Manuel Castells (Castells, 2000a; 2000b), Saskia Sassen (Sassen, 2005a, b; 2002) and Nigel Thrift (2005). This work is more explicitly concerned than the approach developed in this study with the transformative potential of ICTs and its connection to globalization and new models of production and distribution.

The work of these three theorists originates from different fields. Sassen's work is more clearly sociological whereas Nigel Thrift's arguments are rooted in geography. Manuel Castells combines sociological and geographical perspectives, but with a stronger empirical focus than the other two. Nevertheless, they all share an intellectual agenda concerned with the specific influence of ICTs on socio-economic dimensions of globalization; the way ICT infrastructures become embedded in different organizational and institutional contexts; and the association of these developments with new processes of production, distribution and the evolution of labour markets. These include the increased involvement of users in the production and distribution of informational goods (Cooper, March 2005; Currah, 2006) and the evolution of highly-skilled, technical labour markets, through, for example, the growth of contingent labour (Kunda, et al., 2002). Another common characteristic in the work of these authors concerns the wide scope of enquiry. All three focus on socio-economic processes of change and transformation informed by a distinctive theoretical approach. It is important to acknowledge these contributions in positioning the research findings of this study in the wider context of the socio-economic changes accompanying the spread of ICTs, and in indicating future directions for research.

In developing insights into the interdependence of the gift and exchange economies and the way power relations are embedded and articulated through the social organization of learning and the division of labour, this research has developed a topology that takes into account dynamics that, at first glance, appear exclusively external (processes of commodification) or internal (dynamics of collaboration) to the organization of F/OS projects. This was necessary in order to specify not only how F/OS communities are similar to other socially grounded processes of production and distribution, such as peer-to-peer networks, but also how they differ from them. These aspects are often neglected in the broader contexts of investigation adopted by Sassen, Castells and Thrift. Similarly, although this research has highlighted issues such as how resources are distributed and managed, the conceptual framework (see Chapter 2, section 2.6) made it feasible to position such considerations within the interdependencies that form between different actors at different levels of interaction.

7.5 *Conclusion*

This chapter has outlined the main findings of the research and highlighted the structural and discursive dynamics of participation and collaboration in F/OS projects. Structural relates to ordering the characteristics of collaboration and work, such as those imposed by the division of labour and employment relations. These dynamics were highlighted in the context of the first-level analysis of the findings, which utilized Foucauldian notions and methods, but which also drew on a strand of work in Economic Sociology, specifically the idea of structural embeddedness (Granovetter, 1985; Uzzi, 1996). This work focuses on the way that cooperative and social ties developing in networks of material exchange influence economic activities. The term discursive refers to frames of reference and types of rationalization associated with specific knowledge domains, such as the definition of meritocracy as technocracy, and the ritual enactment of the idea of community. These aspects of participation have been captured by the notion of technologies of communities, extending an idea proposed by Nikolas Rose (1999). In particular, the first-level analysis of the findings indicates that the gift economy is embedded in the exchange economy; that the boundaries between the gift and exchange economies are permeable and that their respective needs and requirements are constantly being negotiated in the context of projects; that legitimate peripheral participation is shaped by the demands of production; and that programming and non-programming teams are distinctive in terms of their priorities, make-up and rhythms of participation.

The first-level analysis suggests that centrality and peripherality are associated with the division of labour, which, in turn, is grounded in employment relations and frameworks of institutional support. It also suggests that, although F/OS is grounded in the gift economy, it is also part of the demands and dynamics of the exchange economy. These findings contribute to the development of *an improved understanding of how the gift economy is embedded in the exchange economy*. The point of departure from studies in one of the traditions of Economic Sociology is that this research examines the embeddedness not of the social in the economic, but of the economic in the social. This approach balances the oversocialized view of F/OS communities often adopted in the literature and contributes to a better conceptualization of the notion of structural embeddedness, that is frequently underspecified (Uzzi, 2001).

This approach enables *a deeper understanding of the role of peripheral contributors and of peripheral development in F/OS projects*. The term 'autonomous peripherality' is used to describe the activities and roles of non-coders, a group neglected in the F/OS literature. The broadening of the focus of investigation to take account of different groups of contributors and types of contributions has important methodological implications for the study of the dynamics of collaboration in other communities focusing on the production of complex goods.

Learning is presented in this study as an important factor in controlling access and participation that is also closely linked to the process of production. This insight forms the basis of a critique of learning as an enculturation process that is present in many studies that employ or adopt a CoP perspective.

The second-level analysis of the findings highlighted three aspects of *technologies of communities and developed the notion of a relational meshwork*. Foucauldian ideas were drawn on predominantly but Goffman's notion of face-work was exploited to describe and analyse *the programme of meritocracy* in which the definition of meritocracy as technocracy and its decoupling from institutional frameworks of support together with its grounding in social relations, establish a hierarchy that is essential for maintaining the basis of participation and for attracting new contributors to F/OS projects.

The conceptual framework also provides a basis for insights into *the ritual and strategic enactment of the idea of community*. The results suggest that team and personal performances utilizing the notion of community are essential for mobilizing volunteer resources and for reifying the conception of F/OS projects as autonomous spaces.

In addition, the tools and techniques developed to address the emergence of a set of problems specific to F/OS community management and governance were discussed. This technology constitutes communities as objects and subjects of governance.

Broadly speaking, technologies of communities can be said to constitute programmes of action, whose realization and affirmation need to be constantly renegotiated against the structural and discursive tensions introduced through the dynamics of commercialization and peripheral participation. As Foucault points out:

> These programs don't take effect in institutions in a integral way; they are simplified, or some are chosen and not others; and things never work out as planned. But what I wanted to show is that this difference is not one between the purity of the ideal and the disorderly impurity of the real, but that in fact there are different strategies that are mutually opposed, composed and superposed so as to produce permanent and solid effects that can be perfectly understood in terms of their rationality, even though they do not conform to their initial programming: this is what gives the resulting apparatus its solidity and suppleness. (2000a: 231-232)

In order to conceptualize the 'resulting apparatus' in this study the concept of relational meshwork was introduced. The relational meshwork describes the interweaving of technologies of communities with the discursive, structuring and structural forces introduced by the dynamics of development such as commercialization and peripheral participation. Sometimes these forces are oppositional. For instance, the idea of community as a unified, homogenous space is undermined by the hybrid view of community that emerges when the agendas and priorities of different groups are taken into account. In some cases the relation between them is one of reification. For instance, the establishment of the community as a subject of governance can be seen to be consistent with the demands of the exchange economy.

The constitutive forces and technologies of meshwork, which can be visualized as nodes, perform multiple functions. They are linked by different threads - different relations, that can be either reifying or oppositional in nature. Moreover, the interplay between these different logics and practices creates different opportunities for agency and co-option.

The analysis in this thesis contributes to understanding of the redefinition of the relationship between sociality and economic production represented by community-led F/OS projects and they way this evolves around the construction and appropriation of communities as separate socio-economic spaces with unique production capabilities.

8 Conclusions

8.1 Chapter overview

This chapter recapitulates the main theoretical, methodological and empirical contributions of the study and situates them in the context of other relevant work, highlighting the limitations of the study and proposing avenues for future research.

The study provides an innovative perspective on open collaborative communities in the specific context of F/OS projects. In setting up a framework for the analysis three interrelated challenges had to be confronted. Firstly, two under-examined aspects of collaboration and participation, commercialization and peripherality had to be examined in parallel. Secondly, a framework was needed that would highlight both the structural and discursive tensions that these two dynamics introduce in the F/OS development process. One that would illuminate community life and help explain the persistence of the modes of rationality implicit in the technologies of communities. Thirdly, a framework was needed capable of supporting a nuanced account of the opportunities for agency and control that arise in the space of community-driven F/OS projects which embrace complex, and often ambiguous, power relations.

In this chapter section 8.2 reiterates the main theoretical, methodological and empirical contributions of the research and situates them within the context of the F/OS literature. Section 8.3 discusses the limitations of the study and the generalizability of the findings. Section 8.4 outlines the short and long-term goals of a future programme of research that builds upon the insights from this study. Section 8.5 summarizes the key points of this chapter.

8.2 Theoretical, methodological and empirical contributions

This section outlines the main theoretical and methodological advances contributed by this study and summarises the main empirical findings and comments on their generalizability. The study offers a deeper understanding of some of the configurations of power supported by ICTs and some of the qualitative transformations that may be associated with the emerging information society. The results also contribute to theoretical and methodological discussions on the organization of online communities that focus on the production of complex goods. It does so by developing the notion of peripherality and drawing attention to the complex division of labour in the production of these goods. It is demonstrated that rather than homogenous communities, large-scale F/OS projects are better envisaged as constellations of practice that are embedded in existing socio-economic and institutional structures. The communities of practice perspective is extended in this study by a discussion of how learning relations "are enabled and constrained by their embeddedness in relations of power" (Contu and Willmott, 2003:283). By drawing attention to the multiple connections of F/OS communities with existing socio-economic networks the results suggest that the gift economy is embedded in the exchange economy, and provide a basis for a further elaboration of the notion of *embeddedness.*

The main theoretical contribution of this study is in blending together the insights from the CoP perspective, Foucault's ideas on power, and Goffman's notion of face-work, to construct a conceptual framework that enables the structure and dynamics underlying F/OS to be investigated in way that provides insights into key aspects of access, control and power. The analysis of the data in this study has yielded new concepts, including the notion of autonomous peripherality and the notion of 'technologies of communities', and 'meshwork' relationships. These were elaborated in Chapters 5 and 7.

The concept of *technologies of communities* refers to the tools and processes that constitute specific programmes of action which confer concrete forms to specific rationalities. In section 7.3, Chapter 7, three technologies of communities relevant to F/OS communities were highlighted and discussed: the programme of meritocracy, the ritual and strategic enactment of the idea of community, and the tools and techniques developed to address a set of problems specific to community management and governance.

The programme of meritocracy concerns the persistence of the meritocratic discourse and specifically the definition of meritocracy as technocracy as the prominent mode for rationalizing social order and hierarchy within F/OS projects.

The strategic and ritual enactment of the idea of community concerns the invocation of the idea of community through individual and team performances, that appeal to commonly held values and ideas. These performances are essential for establishing and reifying the idea of a coherent community. They are also indispensable for mobilizing volunteer resources, which due to the proliferation of different development agendas, can no longer be taken for granted.

The tools and techniques developed to address a set of problems specific to community management and governance. This technology relates to the processes of objectification and, more specifically, to the tools required to vizualize, survey and map community space and the techniques developed to manage labour. The knowledge generated through the study of involvement, or involvement in these processes, is increasingly formalized as part of the effort to cultivate and manage volunteer communities.

Other studies of F/OS seem to take for granted the modes of rationality implicit in 'technologies of communities'. For example, many accounts adopt the view that a person's position in the project hierarchy is a function of his or her contributions. Some simply accept the ideological, communal basis of participation. A considerable number of studies consider the technical tools and leading project contributors as decisive factors in the success of F/OS software. The acceptance of these modes of rationality is pervasive in studies of F/OS development both theoretical and empirical. For example, as indicated in Chapter 1, section 1.3.2, many Social Network Analysis studies assume that the basis of F/OS project organization can be almost exclusively attributed to internal project dynamics. *In this study a theoretical and methodological framework has been developed which enables the parallel examination of structural and discursive tensions associated with commercialization and peripherality. on the one hand, and 'technologies of communities', on the other.*

One of the main insights afforded by this study is that *technologies of communities can never fully be realized and that they are constituted despite the obvious discursive and structural tensions introduced by commercialization and peripherality.* As indicated in Chapter 7, section 7.3 one of the reasons for their persistence lies in their multiple connections with existing discourses operating across different institutional domains, such as the increasing importance attributed to the notion of community in politics, science and the workplace. Another reason for their persistence may lie in their importance for constituting F/OS communities as objects and subjects of governance *(tools and techniques for community governance and management)*, in safeguarding an equitable basis for participation *(programme of meritocracy)* and in mobilizing community resources and reifying the idea of a coherent community, despite the divergent agendas and priorities of different groups within the project *(ritual and strategic enactment of the idea of community)*.

The interweaving of technologies of communities with the dynamics of commercialization and peripherality has been seen as shaping a complex topology of participation or a 'relational meshwork' (see Chapter 7, section 3). The threads of this relational meshwork consist of structural and discursive forces that can stand in opposition to or reify one another. For instance, the idea of community as a unified, homogenous space is undermined by the hybrid view of community that emerges when the agendas and priorities of different groups are taken into account. One the other hand, the establishment of the community as a subject and object of governance can be seen as consistent with the demands of the exchange economy. These relationships have been shown to create different opportunities for control and agency and to be indicative of the redefinition of the relationship between sociality and economic production that supports community-led F/OS development.

The examination of commercialization involved the investigation of different aspects of the relation between the gift and exchange economies. This included examination of the role of paid developers, the GNOME Bounty Hunt contest, a new type of interface developed between companies and communities, and a detailed investigation of the dynamics that underlie the initial stages of commercialization in a volunteer-driven F/OS project, encapsulated by the Gstreamer/Fluendo case.

In Chapter 4 a typology of paid programmers was developed based on the types of their mandates, their sponsorship arrangements and their community ties. Two specific categories of developers were examined in detail: those with no prior community connections ('outsiders') and those with close community ties ('insiders'). The investigation of the case of outsiders revealed the progress that has been made in accommodating corporate needs, such as the regularization of release cycle, and highlighted some persistent disconnections, such as the different processes for quality assurance and measuring performance employed by companies and communities. The analysis also indicated that, in general, 'outsiders', are not treated any differently from other new contributors. Their contributions are judged on the same basis as those of other developers and they are given the same priority in terms of assistance and attention as other first entry programmers.

The most interesting insights were afforded by the examination of the role of 'insiders'. This group includes programmers who are hired on account of their volunteer contributions and 'outsiders' who gradually develop their community ties. The qualitative findings indicated that these individuals play a leading role in projects and are endowed with a legitimacy that cannot easily be questioned. For their part 'insiders' have an increased sensitivity with regard to community-corporate relations. The results of the quantitative findings confirm the importance of their role and highlight their different patterns of participation and contribution as compared to volunteers. Specifically, the analysis indicates that there is a strong association between employment and maintenance of critical, infrastructural modules and attendance at community events.

The case of the corporate sponsored Bounty Hunt revealed some of the tensions associated with the different perceptions of the way corporate needs should be accommodated within the structure of projects and emphasizes the question of how the development agenda of F/OS project development is formulated. The cases of the Fluendo SL company and the Gstreamer project revealed some of the complications arising in the transition from a volunteer project to one influenced by the presence of a corporate player, and staffed by a number of key community contributors. For example, the faster pace of development introduced by Fluendo SL made it difficult for volunteers to absorb and keep up with the changes and raised the barriers to participation for new developers. In addition, the Gstreamer/Fluendo case suggests that despite the influx of new resources, the blurring of the boundaries between the corporate and the communal seemed to work in favour of companies more than communities. It was hypothesized that the investment of knowledge in corporate entities and paid employees raises significant questions about the future of the projects to which they contribute. Investigation of this hypothesis would involve long-term examination of the dynamics of collaboration between corporate actors and communities which is beyond the scope of this study.

How do these findings compare with the insights and results from similar F/OS studies? As indicated in Chapter 1 a growing number of studies has questioned the idea of the decentralized character of F/OS development by drawing attention to the existence of a core group of developers. This has been demonstrated both within the context of development and in terms of communication patterns. Moon and Sproul (2002), for example, drew attention to the highly skewed distribution of traffic on the Linux mailing lists and argued that interactions in this project are centred around a few individuals. Koch and Schneider (2002) indicated that the bulk of the code contributions in GNOME was undertaken by a core group of 11 developers. These studies essentially argue that the social structure of F/OS project involves a core-periphery model of organization. Parallel to this line of investigation is another group of studies that is predominantly concerned with questions of motivation. This work indicates that economic incentives, either in the form of direct remuneration or in the form of access to venture capital and global corporate networks, play an important role in F/OS development.

This study complements both groups of studies in indicating how employment affects the division of labour within the context of specific F/OS projects. This challenges the idea that the social structure of F/OS communities is entirely dependent on internal project dynamics, since the ability to work on a project on a full-time basis inevitably affects the quality and quantity of individual contributions. In addition, the composite examination of the dynamics of commercialization enabled the formulation of a more nuanced view of the relation between the exchange and gift economies. This takes into account their complex interdependencies at different levels of interaction and development. The fact, for example, that many employed developers maintain modules they are not directly paid to work on indicates the limitations to examining this relationship in terms of individual motives. This approach is quite different from the examination of F/OS software commercialization at the institutional level in looking at emerging organizational structures and licencing schemes, and from a business point of view because it provides insights into how the commodification process affects development at the meso and micro levels.

These findings contribute to the further elaboration of the notion of embeddedness and a deeper understanding of the dependencies that are developing between the gift and exchange economies within the context of F/OS. As indicated in Chapter 7 studies on embeddedness within the tradition of Economic Sociology are predominantly concerned with the embeddedness of economic actions in social relations. Brian Uzzi (Uzzi, 2001: 208) defined economic embeddedness as the way: "quality and network architecture of material exchange relationships influence economic activity". This view contrasts with the dominant view of F/OS in which it is the network of social relations that is considered to be the most important issue and where the relation between the gift and exchange economies is depicted as one of balance between extrinsic and intrinsic motivations. In writing about the economics of social production Benkler (2006: 97), who coined the term 'social production' to describe the specificity of the F/OS model of development, says: "The point is simple money-oriented motivations are different from socially-oriented motivations. Sometimes they align. Sometimes they collide. Which of the two will be the case is historically and culturally contingent".

It is interesting to compare Benkler's argument with some of Mauss's concluding remarks in *The Gift:*

> The terms that we have used-present and gift-are not themselves entirely exact. We shall, however, find no others. These concepts of law and economics that it pleases us to contrast: liberty and obligation; liberality, generosity, and luxury, as against savings, interest and utility-it would be good to put them in the melting pot once more. We can only give the merest indications of the subject. Let us choose, for example, the Trobriand Islands. There they still have a complex notion that inspires all the economic acts we have described. Yet this notion is neither that the free, gratuitous rendering of total services, nor that of production and exchange purely interested on what it is useful. It is a sort of hybrid that flourished.
>
> (Mauss, 1954: 93)

One of the recurrent arguments in this study is that with respect to modality of 'social production', *the importance of the exchange economy should not be underestimated and the complexity of its relationship with the gift economy should not be limited to the level of motives.* This relationship has been shown to affect different levels of development and is often ambiguous and contingent. The case of 'insiders', for example, challenges the view of corporate players as external to development and indicates that the connections between companies and projects develop at different levels of involvement and hinge on complex interpersonal dynamics. At the same time, the strong association of the paid developers with the core group of hackers, and of the volunteers with autonomous peripheral contributors, suggests that structural factors bear directly on the social organization of projects.

The examination of the dynamics of peripheral participation involved two different aspects of peripherality. The first aspect of peripherality focused on the *legitimate peripheral participation* of programmers and learning as a factor structuring participation and access from the perspective of newbies and senior developers. The second aspect of peripherality, *'autonomous peripherality'*, examined the participation of non-coders and highlighted their distinctive characteristics and collaboration with programmers.

The investigation of coders' legitimate peripheral participation highlighted three stumbling blocks to participation for potential F/OS contributors. First, there are difficulties associated with the technical aspects and tools of F/OS development, such as the use of CVS. Secondly, there are conceptual difficulties related to understanding the development process and the architecture of the program, how they are set up, how things fit and how they are expected to be put together. Thirdly, there are difficulties related to how newbies situate themselves in the development process, and the selection of tasks that are appropriate to their level of skills. Newbies describe integration as a slow learning process through which they build up their skill sets and community knowledge and position themselves in the development through their choice of tasks.

The analysis suggested that this process is seen differently by experienced developers. The discourse of senior developers relating to new contributors appears to be informed by a production-oriented view of the development process. This shapes their expectations in terms of the behaviour and performance of newbies and guides their decisions about helping them. One of the first things that they mentioned in assessing the potential of new contributors was their chosen entry point into the development, and the way they initially presented themselves on the mailing lists. Ideal candidates are those that demonstrate self-reliance, initiative and commitment either by having choosing a task appropriate to their skill level or by having formulated a general idea about the projects' architecture. According to senior developers peripheral participation is subject to the imperative of production and should take place in the background and not in the foreground of development. In short, as social relations are embedded in economic relations, so learning relations are embedded in the relations of production.

With regard to *legitimate peripheral participation*, this study complements those concerned with the integration and socialization of new developers such as the work of Ducheneaut (2005) and von Krogh et al. (2003). The perspective in this study highlights learning as a structuring factor of participation which is shaped by the demands of production. The view that emerges as a result of the research is that successful joining does not just involve the appropriation of hacker ideals. Becoming a legitimate peripheral member in F/OS involves overcoming specific conceptual and technical difficulties while demonstrating a particular attitude to learning and contributing. This suggests that appropriation of the specific codes of conduct is expected to precede visible involvement.

At a theoretical level this perspective has provided the basis for critiquing the view of learning as an enculturation process that is frequently adopted in most applications of the CoP perspective. As we have seen this view of learning is consistent with the cultural/ideological perspective underlying many F/OS studies and is in line with the ideological/consensual view of CoP. However, in Lave and Wenger's (1991) original formulation of the CoP perspective power relations are conceived as central in enabling and constraining access to positions of initial peripherality and subsequent potential mastery. In this study this underdeveloped aspect of Lave and Wenger's theory has been extended by indicating that learning is shaped by the demands of production, and regulates access to positions of legitimate peripheral participation. The higher barriers to access imposed by the faster rates of development introduced by paid developers are indicative of the way structural characteristics, in this case employment, shape the learning process. At the same time, the difficulties associated with the lack of learning resources, control access to the periphery and to senior developers' attention horizons.

The examination of non-coders' participation indicated *that 'autonomous peripherality' is a distinctive sphere of activity with different aims, priorities and rhythms of participation as compared to those of coders.* Translators, documenters and usability experts place greater emphasis on the values of access and ease of use than coders who prioritize technical excellence. In addition, in their dependence on the work of coders, translators' and documenters' contributions are seasonal in character and culminate during the project's freeze periods. However, despite the fact that a lot of progress has been made in accommodating non-coders needs in the release cycles, unavoidable tensions emerge through the desire of programmers to tinker with their code to the last minute and non-coders' need to maintain the freeze. In proportion to the efforts made to invite more programming contributions, significantly greater efforts are being made to lower the barriers to participation for non-coders.

However, as the results indicate, *coding skills are persistently valued more than non-programming skills.* This bears upon the position of peripheral contributors in the social hierarchy and, to a certain extent, in decision-making. The results of the quantitative analysis indicate that non-coding activities are primarily driven by volunteers. Moreover, in contrast to paid developers hired to work on GNOME and KDE, volunteers do not attend community events with any regularity. Combined with paid developers' concentration on critical, infrastructural modules of the development, this leads to the conclusion that the division of labour in terms of contributors' primary areas of activity is associated with employment status and specifically with employment directly related to projects.

The insights into 'autonomous peripherality' can be viewed in relation to those afforded by studies arguing in favour of a core-periphery structure of F/OS projects. In these studies the periphery is defined on the basis of circumstantial or infrequent programming related contributions, such as small patches (solutions to specific software problems), and bug reporting (reporting of software faults). The contributions of non-coders are not accounted for. In their article 'The social structure of free and open source development', Crowston and Howinson (2005) recognize the need to differentiate between more and less active developers and between active and passive users, but essentially reproduce the dichotomy between developers and users. Moreover, as we have seen, most examinations, especially in the case of F/OS development, are predominantly concerned with the projects' core and co-developers (see Figure 8-1). Recent studies, such as Karim Lakhani's (2006) work on 'The Core and Periphery in Distributed and Self-Organizing Innovation Systems' have begun to address this gap by highlighting the role of peripheral developers in large F/OS projects. Having studied how new software features are introduced and implemented Lakhani concludes that: "peripheral members are responsible for developing a significant majority of functionally novel software features" (Lakhani, 2006: 20). Despite the importance of his contribution he does not take into account the wider context within which F/OS development takes place as indicated by the important role played by non-programmers. Crowston and Howison's study indicates the need to elaborate more complex models of F/OS development that take into account types of contribution other than coding, as shown in Figure 8-1.

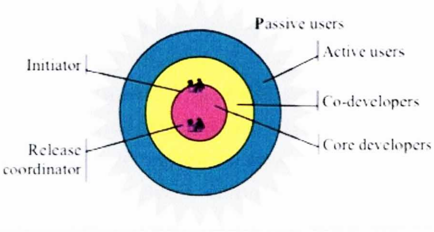


Figure 8-1: A synthesized FLOSS development team structure from Crowston and Howinson, 2005: p. 7

According to the findings of the present study, Crowston and Howison's insights can be amended as shown in Figure 8-2:

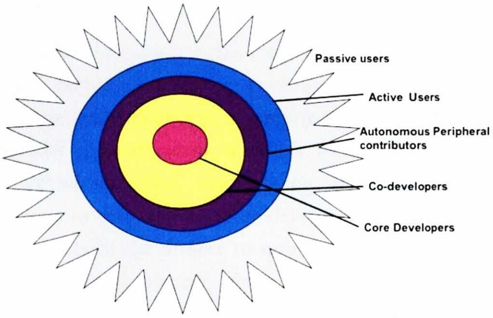


Figure 8-2: Revised version of Crowston and Howick's Figure based on research findings

As indicated in Figure 8-2, the team structure of F/OS development includes an additional group made up of autonomous peripheral developers that is situated between the groups of co-developers and active users.

Throughout this study it was been argued that the notion of the periphery is a particularly understudied area of online communities (see especially Chapter 5, section 5.2). By highlighting different types of contributions, specifying the special characteristics of the organization of teams of translators, documenters and usability experts and their relations with coders, this study has helped to redefine the concept of the periphery in large F/OS communities. The results of the study question the validity of the notion of large F/OS communities as a relatively homogenous CoP, establishing peripherality as a distinct sphere of activity rather than as the preparatory stage in the transition to the community's centre. In particular the results indicate that different groups of contributors operating in the context of large F/Os projects share the same overarching goals, but also have distinctive priorities and agendas.

These findings offer an arguably more accurate conceptualization of the organization and the division of labour in large F/OS communities. However, they pose the broader question of whether the notion of a single 'community' is appropriate for analyzing relations in large F/OS projects. As indicated in Chapter 2 the notion of the 'constellation of practices' was developed by Wenger (1998) in order to describe broad and diffused establishments that incorporate multiple CoP, organized on the basis of geographical, cultural or economic continuities, such as organizations. The specialization and variety of skills other than programming identified in the context of this research and their complex ordering, therefore, raise the question of what type of organization F/OS projects constitute and are evolving towards and what types of conceptual apparatus are appropriate for describing the novel balance between social and production relations underlying F/OS development processes. *This has significant methodological implications as it invites a broadening of the framework of investigation in the future to take into account the diverse and particular characteristics of F/OS projects.*

Another aspect that distinguishes this study from existing mainly quantitative studies that investigate communication or development patterns in F/OS is *the framework focusing on the qualitative character of participants' contributions which takes into account an offline dimension of participation, which in the context of this study refers to community conference attendance.* The meta-coding scheme used to categorize participants' contributions in different areas of development, ranging from the most crucial infrastructural aspects of the project to non-coding, peripheral, activities, was helpful in building an interesting picture of the distribution of volunteer and paid developers. The results concerning conference attendance suggest that offline aspects of F/OS development should be examined more closely.

The empirical findings demonstrate that commercialization and peripheral participation are not independent phenomena but are closely connected and have ambiguous implications for the development of F/OS communities. The commodification of F/OS projects and their increasing use by a large base of non-technical users increases the demand for high quality translations, more user-friendly documentation and better-designed interfaces. The normalization of the release cycle and the standardization of various aspects of development demanded by corporate players was found to facilitate the integration of new volunteers. At the same time, however, the high level of expertise demanded in F/OS projects that are directly tied to corporate products and services appears to raise increasingly the barriers to access for new contributors.

Table 8-1 presents the principal insights afforded by the study organized on the basis of the operationalized research question presented in Chapter 3, section 3.2

	First level operationalization	**Second Level Operationalization**	**Answers/ Results**
Peripherality	How is the idea of the peripherality and centrality experienced, structured and constructed?	What are the frameworks of meaning and experience associated with participation in each sphere, how are different skill sets and contributions evaluated between the examined groups?	*Programmers appear to place greater emphasis on technical excellence, whereas peripheral contributors prioritize access and ease of use. Autonomous peripherality is a distinctive sphere of activity with different aims, priorities and rhythms of participation as compared to those of coders. However, coding skills are persistently valued more than non-programming skills.*
		Is the idea of peripherality negatively constructed?	*Not by autonomous peripheral contributors. However, this group is seen to attain higher status within the community with more difficulty than programmers.*
		Are peripherality and centrality connected with employment status?	*Yes, as indicated by quantitative findings, many core developers are employed to work on the projects, whereas volunteers drive peripheral activities.*
	How does learning structure access and participation?	How does learning structure access and participation?	*By the difficulties newbies have to face when joining a project (see below) and by the requirement of productivity. According to senior developers learning is an activity that needs to take place at the background and not the foreground of development.*
		What are the barriers to access for new contributors?	*There are three different sets of difficulties: a) the technical aspects and tools of F/OS development, b) conceptual difficulties related with the development process and the project architecture, c) situating oneself in the development process and selecting appropriate tasks.*
		How is learning connected with other fundamental project dynamics (i.e. meritocracy, volunteer basis of participation, gift culture)?	*Projects make significant efforts to lower the barriers to participation and attract volunteer contributions. However, commercialization introduces a faster rate of development and demands more rigorous processes of quality control, which raise the barriers to entry.*
↓↑	How are relations and practices between different groups of contributors organized and maintained?	How is cooperation between coders and non-coders structured?	*A lot of progress has been made in accommodating the needs of non-programmers, through the establishment of freezes, non active phases for programming, in the development process. The established pattern of development means that the contributions made by translators and documenters are seasonal in character. However, there persists an inherent tension between accommodating the needs of peripheral contributors and the desire of coders to tinker with their code until the last minute.*
		How are the different group agendas, including those of paid developers, prioritized?	*The needs of coders remain more important than those of non-programmers. The requirements of 'outsiders', paid developers appear to be negotiated on the same basis as those of volunteers.*

			However, integrated paid developers appear to be in a more advantageous position to achieve their aims, due to their prominent community positions and the fact that they can spend more time on development than volunteers.
	How is the idea of community contested, confirmed and perceived among the various groups of contributors?	*Despite the different agendas operating within the context of projects and the structural and discursive tensions introduced by peripherality and commercialization, the cohesion of the community is affirmed with the help of three technologies of communities. The program of meritocracy safeguards the equitable basis for participation, the tools and techniques for community governance and management constitute F/OS communities as objects and subjects of governance and the ritual and strategic enactment of the idea of community help mobilize volunteer resources and reify the idea of a coherent community.*	
Commercialization	What is the relation between the gift and exchange economies? What impact does commercialization have on development and community dynamics?	How are the boundaries between companies and the communities managed and negotiated at the meso and micro level?	*The boundaries between companies and communities are permeable and their respective needs and requirements are constantly negotiated within projects. The connections between companies and F/OS projects develop at different levels of involvement and hinge upon complex interpersonal dynamics. As indicated by the role of community integrated paid developers and bounty hunt contests, companies cannot be considered as external to development, but have a direct impact on project dynamics and agenda.*
		What is the role of paid developers in development and community life?	*'Insiders', developers with close community ties, play an important role in development and community life. They contribute heavily and maintain critical aspects of the code, which are more technically challenging, and participate more than volunteers in community events.*
		How do new types of interfaces set between the community and companies, such as bounty hunt contests influence project dynamics?	*They reveal some of the tensions associated with the different perceptions of the way corporate needs should be accommodated within the structure of projects and emphasized the question of how F/OS projects development agenda is formulated.*
		What areas do paid developers contribute to and maintain?	*Paid developers work primarily on infrastructural aspects of development. However, they appear to be also involved and maintain modules that they are not directly paid to contribute to.*

Table 8-1: Operationalized questions and related findings

The generalizability of the findings from this study is limited to some extent by the methodological choices discussed in Chapter 3 and the limitations outlined in the section below. The technical and social characteristics of the GNOME and KDE projects, which constituted the primary case studies, mean that it is important to be cautious in drawing out wider implications from the study. Both these projects are community-led and community initiated, socially and technically mature F/OS projects that comprise different levels of development. They provide an easy-to-use GUI environment, a host of applications, ranging from office suites to email clients and MP3 players, and include a collection of tools and programs that are being used to develop further the GUI environment and the user applications. The fact that these projects address different types of audiences, both technical and non-technical, may mean that they provide a wider scope for autonomous peripheral development than more intensively technically oriented F/OS projects, such as Apache server or the Python computer language, would allow for. In Chapter 3 it was argued, however, that the conclusions drawn from examining the GNOME and KDE projects are relevant when the long-term course of F/OS development is taken into account. The fact that F/OS begins to address the needs of a wider and more divergent user base in conjunction with the application of F/OS principles of collaboration in other domains of production, such as content, means that autonomous peripheral participation may become an increasingly important aspect of participation.

In addition to the specificity of the particular knowledge domain addressed by F/OS projects, there appear to be a number of other socio-economic, historical and development factors that shape the dynamics of collaboration and participation, influence the balance between the gift and exchange economies and set the conditions for peripheral participation.

The insights that have emerged as a result of this study included the following:

- *Knowledge domain* new insights into the audiences addressed and the types of expertise involved in F/OS development.
- *The degree of social and technical maturity*: whether there is a well-established community of volunteer developers built around the project and the implications of the project release cycles.
- *Involvement and barriers to entry for new developers:* the ease of participation in the development process and the efforts being made to facilitate the integration of new contributors.
- *Adoption:* Whether the code base is used widely and whether it is used by corporate or public organizations in a standalone fashion or as the basis for new products and services.
- *Institutional context of development*: Whether F/OS projects constitute a community or a corporate initiative and what the licensing terms are within which the code is made available.
- *Governance:* Whether the project has a clearly defined governance structure and the form that this assumes.

The variety of institutions sponsoring developers in F/OS projects indicates that generalizations regarding the structure and organizing principles of F/OS should be qualified pending the development of more refined models of the organizations, projects and institutions that populate the F/OS space and the way that they relate to each other. In the course of analyzing the quantitative findings, some of the complexity of the layers of institutional involvement and support underlying F/OS development were revealed. As indicated in Chapter 6 the process of identifying the projects' maintainers showed that many of them seemed to be affiliated, either as students or employees, with a variety of educational and research institutions. In the context of this study affiliation was defined as employment by an organization directly involved in F/OS development, and therefore this type of institutional support could not be taken into account. The point is that volunteers do not operate in an institutional void. Sometimes developers will work on F/OS projects during their working hours without the knowledge of their employers. There is always some kind of support in place that allows volunteers to do their work.

8.3 Limitations of the study

This section reflects on the limitations of this study. The discussion of the research limitations provides a starting point for elaborating a broader research agenda for studying power relations in open, collaborative communities, focusing on the production and distribution of complex goods.

One limitation of this study is that it focuses on internal aspects of collaboration and regards F/OS projects as the primary loci of power. This decision was taken in the light of the aim which was to examine the complexity of the institutional space for F/OS development. This meant that intra community aspects of collaboration were not examined and the broader influence of emerging F/OS institutions, such as those concerned with the promotion of specific standards, were not taken into account. Many F/OS developers work on several different projects. An analysis of the way they distribute their effort between projects or the way they move from one project to the next in combination with their career paths, would have provided additional clues as to the complex dynamics that underlies the evolution of F/OS.

This relates to another limitation of the study, that is, its static character. The research design provides only a snapshot of the two projects' development. In some cases, for example, the Gstreamer/Fluendo nested case study (see Chapter 4, section 4.4), the research examined a transitional aspect of the projects' socio-technical development. Although the extensive phase of data collection, which lasted two years, gave a sense of project and individual progression, the timeframe was limited and the study would have benefited from a more dynamic examination of certain issues. One of the questions that emerged from the quantitative analysis, for example, concerned the relationship between involvement in critical aspects of development and employment, that is, whether companies hire developers who are already on a 'critical path' in the projects' development or whether programmers become core developers given the opportunities for increased participation offered through their employment. This question can only be examined dynamically, by investigating the careers of paid and core developers as they evolve over time. An examination of projects over a broader timeframe could also help to substantiate claims made in this study with respect to autonomous peripherality and whether autonomous peripherality becomes more significant as the code base matures and reaches a wider user-base.

The study also concentrates only on successful community initiated F/OS projects. However, there are a considerable number of F/OS projects that fail to take off or to gather a thriving community around them. In addition, a growing number of F/OS projects is initiated by corporations and public institutions. A research design that included failed and non-community initiated projects was beyond the scope of this research, but would have provided a basis for comparison of the results reported in this thesis.

During the course of the research there were opportunities to become directly involved in projects, by contributing, for example, in the Greek translations of GNOME. Due to time constraints, however, such opportunities were not pursued. In retrospect involvement at this level would perhaps have deepened my understanding of the dynamics of peripherality and would have improved my access to the projects. Although the study has benefited significantly from F/OS contributors' willingness to explain their views and share their insights, such projects have different levels of access and it is frequently difficult, as an external observer, to gain insights into the inner workings of communities.

The genealogical aspects of the analysis (see Chapter 2, section 2.3.2) were not pursued to the extent that was initially envisaged, mainly because of the need to complete Phase 3 of the study (quantitative examination). For example, the issue of appropriation of managerial techniques developed for the workplace by F/OS developers and the migration of 'community management' know-how back into a formal managerial context could have been further pursued.

One of the weaknesses of the quantitative examination of the patterns of contribution, presented in Chapter 6, concerns the self-reported character of the areas of contribution. It is possible that some aspects of contribution were under or over-reported by certain groups in the survey. For example, not all core contributors may have reported all their aspects of involvement in peripheral activities with the same degree of precision as individuals that contributed only to peripheral aspects of development. In the future this could be addressed in part by quantifying the contributions using CVS statistics. The problem in this case is that not all contributions, especially those related to non-coding activities, have a CVS entry. Another choice with potentially important analytical implications concerns the decision to focus on the primary areas of contribution. This might have undervalued the importance of individuals whose activities span peripheral and core communities.

In addition, the informal character of maintainership creates substantial difficulties in identifying clear patterns of ownership and change. Often, there is no official hand-over of maintainership or hand-over, especially in the case of less important modules, may not be announced in the relevant mailing lists or on project websites. A developer may assume maintainership of a module on an informal basis.

Another limitation of the quantitative study relates to the analytical techniques. As shown in Chapter 6 the quantitative analysis employed mainly descriptive and non-parametric statistical techniques, mostly crosstabulations and chi-square tests. Although these techniques were suitable for answering the specific hypotheses that emerged from the qualitative phase of the study, other statistical methods, such as Social Network Analysis techniques could be used in the future to further explore the data. The application of Social Network Analysis would have helped to visualize the different groups and cliques that organize themselves around the same aspects of development and to indicate their relations to each other. This would have helped to build a more accurate picture of the way different development teams are organized in the space of projects. In addition, it would have been interesting to compare how this topology of participation relates to the location of the GNOME Foundation and KDE e.V. members. This information was gathered by the survey, but was not analyzed because of the scale of effort needed to do so and the need to bring closure to the empirical phase of the research.

The broader issue connected with the study concerns the specific types of dynamics and groups examined. As indicated in Chapter 2 the examination of the dynamics of commercialization and peripheral participation was theoretically driven and aimed to investigate two significant blind spots concerning power in F/OS. However, although they are extremely important, these two dynamics are not the only threads in the meshwork of power underlying F/OS development. As was revealed through the research process, the dynamics related to gender, culture, geography and language are also important threads and could form separate objects of investigation.

8.4 Avenues for future research

This study offers considerable scope for further analysis. For instance, analysis of the relationship between the gift and exchange economies could be enriched by a consideration of the different perceptions of time that underlie these economic models. In future work the seasonal character of peripheral contributors' participation, the cyclical character of the releases, the perception of F/OS projects as continuous work-in-progress could be contrasted with a linear, progressive sense of time, punctuated by the hard deadlines that characterize the exchange economy. Leyshon (2003) examined these issues in the context of peer-to-peer networks and the different time-forms underlying the gift and exchange economies was investigated by Bourdieu (1977) in 'A Theory of Practice'.

With regard to peripherality, the construction of expertise between and within different communities of practice could be further developed in future work.

These lines of enquiry are compatible with the conceptual framework in this thesis, but because of the need to bound the analysis, they were not pursued.

Commercialization and peripherality constitute only two of the threads in the 'meshwork of power' underlying F/OS development. As in so many aspects of social life, race, gender, geography, language and culture, underlie many power dynamics in F/OS development. For example, according to a F/OS developer working for Sun Microsystems, the Indian programmers working for Wipro Ltd, who were used to more hierarchical modes of development and traditional apprenticeship relations than those applying in the context of community development, were often reluctant to take the initiative. A Chilean developer I met at GUADEC admitted in conversation that many of his colleagues were reluctant to contribute to IRC (Interlay Relay Chat) channels because they were not confident about their English language skills. The question of gender also is often an issue among F/OS developers, but it has never been examined systematically.

This study provides many indications of research directions needed to develop a more comprehensive, holistic picture of F/OS community life and development and to validate the concepts of 'technologies of communities' and 'relational meshwork'. A further elaboration of the results of this study should be undertaken. Subsequent research is needed to develop a methodology that will help to triangulate information across various public data sources in order to build a more comprehensive idea of the structure and the dynamics of collaboration in F/OS communities. Such a framework, for example, would allow information yielded by the F/OS mailing lists to be cross-referenced with developers' contributions and yield insights into the short and long-term patterns of activity and stratification across communication and software development.

Another avenue for further research would be to study co-membership across different types of networks and teams within a project. A contributor to GNOME for instance may be a member of the Board of Directors of the GNOME Foundation, a member of the release team, a core contributor and a company's employee. Such an approach would help to align different aspects of community involvement and professional activity.

In future research further elaboration of the relationship between the gift and exchange economies would benefit from an examination of the following issues:

The temporal forms of the exchange and the gift economies within the context of development and the way they might be linked to different perceptions of work, institutional requirements and offline dimensions of cooperation.

In this study some of the aspects of this relation are examined, including the cyclical, seasonal aspect of F/OS development that appear to be underpinned by the conception of programs as perpetual works in progress compared to the more linear nature of commercial exploitation driven by deadlines and the promise to deliver. It would be interesting, therefore, to examine more closely the commercial and F/OS cycles of developments and the way they are informed by similar or different constraints, and their synchronicities or asynchronicities. For example, the requirements of legal departments constrain some commercial developers. Sometimes the process of validating the legal framework of the incorporated code introduces delays in commercial releases. This was the case in the adoption of the Gstreamer code base by Nokia's multimedia Maemo platform. The lengthy legal validation process involved, resulted in the commercial product running an outdated, buggier version of Gstreamer code. Equally important might be the temporal dimensions of community life that appear to be punctuated by a host of opportunities for gathering and celebration, that is, major community events and their impact on the development process. During this study, it was observed that these offline events spur significant development activities. It appears that the opportunity for physically working together and for reconfirming the bonds of cooperation has been underestimated within the context of F/OS research. It would be interesting to map not only major global community events, but also more regionally focused events, and relate them to the development process.

The career, working arrangements and rhythms of participation of community-integrated paid developers.

A major issue is the relationship between employment and attainment of core developer status. Given the importance of F/OS related employment among core developers it would be interesting to examine the career development of these individuals, their working arrangements and the relation between their paid work and their F/OS contributions. This study has shown that such individuals also work on F/OS in their free-time. It would be interesting to compare this line of research with Richard Akerloff's (1982) ideas about labour contracts as 'partial gift exchanges'.

The changing relation between the gift and exchange economies that F/OS communities might represent and the differences in power relations underlying different types of peer-to-peer production and distribution communities.

As indicated by Offer (1997) and Titmuss (1997) reciprocal exchange is a persistent trait in social and economic life. One of the main arguments in this present study is that F/OS communities are indicative of a redefinition of the relationship between sociality and economic production. A more rigorous investigation of this claim would involve historically situating it by drawing upon studies such as that by Carrier (1995) *Gifts and Commodities: exchange and western capitalism since 1700*, and elaborating on the different types of gift-giving and the relations of production that take place across different communities of peer-to-peer production and distribution. On the basis of the present study it is important not to make very broad claims regarding the dynamics of cooperation across fundamentally different communities, such as those focusing on the production of software, content, or the dissemination of movies and music over the Web. Future research will be needed to address the question of the different dynamics of collaboration and participation and the associated relationships of production that characterize different types of peer-to-peer communities.

This relates to another argument made within the context of this study, that is, the need *for elaboration of the notion and role of peripherality across different peer-to-peer communities.*

As a subject for future research this would build on the insight that communities focused around the production, and possibly the distribution of complex and valuable goods, need to be conceptualized as 'constellation of practices' rather than as homogenous communities of practice in order to further elaborate the notion of peripherality. Examination of the differences and similarities between experiences of participation at the centre and at the periphery across different types of communities would add to the understanding of issues of access and the division of labour. Foucault's (1980) insights into the construction of expertise and the subjugation of certain types of knowledge to others will be valuable in pursuing this line of research.

Future research is also needed to investigate the validity of two key concepts employed in this study: those of 'technologies of communities' and 'relational meshwork'. An examination of the dynamics of participation across different communities of production and dissemination would contribute towards this goal by providing additional empirical insights. However, more theoretical work is also needed to relate these concepts to other lines of research that attempt to provide a coherent theoretical framework to disentangle the web of discourses, practices and power relations that inform the relationships between the digital, the social and the material aspects and the potentially transformative character of the migration of certain aspects of sociality and the economy to the Internet. In this case, it would be helpful to consider the concept of mediation as developed by media and communication scholars such as (Meyrowitz, 1985; Silverstone, 2005; Thompson, 1995) in extending these concepts theoretically.

8.5 Concluding remarks

In this chapter the key insights afforded by this study have been recapitulated, and their significance highlighted in the context of comparable studies, to assess their generalizability in view of the limitations of the study and outline a future programme of research.

The results of the study have improved our understanding of some of the qualitative transformations that many have come to associate with the emerging information society. In particular, this study provides a basis for a critical assessment of the claims made in relation to the innovative character of F/OS communities by examining two blind spots in our understanding of the way power relations operate in large, community-led F/OS communities, commercialization and peripheral participation. The examination of these two dynamics of collaboration and participation has provided a basis for challenging the idea that the organization of these communities is an emergent phenomenon that depends mainly on internal project dynamics. The research findings indicate that learning relations are embedded in the relations of production, and social relations are embedded in economic relations. The results are helpful in balancing oversocialized conceptions of F/OS organization by drawing attention to the structural dynamics underlying participation in F/OS communities, showing that these are established through the complex division of labour, which, in turn, is partially grounded in employment relations. The results also highlight the continuity in change and the change in continuity by drawing attention to the way existing forms of rationality and their associated programmes of action, encapsulated by the notion of technologies of communities have been transformed within the context of F/OS and are essential to the dynamics of participation. This study establishes a good foundation for challenging new lines of research that more closely examine the transformation of the relationship between sociality and production supported by ICTs.

Bibliography

Aigrain, P. (2002). 'A Framework for Understanding the Impact of GPL Copylefting vs. non Copylefting Licenses.' Online paper available at: http://opensource.mit.edu/papers/aigrain2, last accessed 07/08/04.

Akerloff, G. A. (1982). 'Labor Contracts as Partial Gift Exchange.' The Quarterly Journal of Economics 97, no. 4: pp.543-569.

Allen, J. (2003). Lost Geographies of Power. Malden VA: Blackwell Publishing.

Appadurai, A. (1990). 'Disjuncture and Difference in the Global Cultural Economy.' In Global Culture, edited by Featherstone, M., pp.295-310. London: Sage Publications.

Baldwin, C. and K. Clark (2004). 'The Architecture of Cooperation: How Code Architecture Mitigates Free Riding in the Open Source Development Model.' Available at http://opensource.mit.edu/papers/baldwinclark.pdf, last accessed on 23/07/2003.

--- (2005). 'The Architecture of Participation: Does Code Architecture Mitigates Free Riding in the Open Source Development Model?' Harvard Business School Working Paper, Final Version.

Barbrook, R. (1998). 'The High-Tech Gift Economy.' First Monday: http://www.firstmonday.org/issues/issue3_12/barbrook, 3, no 12, last accessed 12/12/2005

Bauer, A. and M. Pizka (2003). 'The Contribution of Free software to Software Evolution.' Online paper available at: http://opensource.mit.edu/papers/bauerpizka.pdf, last accessed 09/08/04.

Behlendorf, B. (1999). 'Open Source as a Business Strategy.' In Open Sources: Voices from the Open Source Revolution, edited by Dibona, C., S. Ockman and M. Stone, pp.149-170. London: O'Reilly.

Benkler, Y. (2002). 'Coase's Penguin, or, Linux and the Nature of the Firm.' Yale Law Journal 112: pp.369-446.

--- (2006). The Wealth of Networks: How Social Production Transforms Markets and Freedom. New Haven, CT: Yale University Press.

Berguist, M. and J. Ljungberg (2001). 'The Power of Gifts: Organizing Social Relationships in Open Source Communities.' Information Systems Journal 11: pp.305-320.

Best, S. and D. Kellner (2000). 'Kevin Kelly's Complexity Theory: The Politics and Ideology of Self-organizing Systems.' Democracy & Nature 6, no. 3: pp.375-399.

Bezroukov, N. (1999a). 'A Second Look at the Cathedral and the Bazaar.' First Monday: http://firstmonday.org/issues/issue4_12/bezroukov/index.html 4, no. 12, last accessed 22/02/05.

— (1999b). 'Open Source Software as a Special Type of Academic Research (Critique of Vulgar Raymondism).' First Monday: http://firstmonday.org/issues/issue4_10/bezroukov/index.html 14, no. 10, last accessed 22/02/05.

Bijker, W. and J. Law, eds (c1992). Shaping technology/buiding society: studies in sociotechnical change. Cambridge, MA: MIT Press.

Bijker, W. E., T. P. Hughes and P. Trevor, eds (1987). The Social Construction of Technological Systems: New Directions in the Sociology and History of Technology. Cambridge, MA: MIT Press.

Bonaccorsi, A. and C. Rossi (2004). 'Altruistic Individuals, Selfish Firms? The structure of motivation in Open Source Software.' First Monday: http://www.firstmonday.dk/issues/issue9_1/bonaccorsi/ , last accessed 7/06/04 1, no. 9.

Bourdieu, P. (1977). Outline of a Theory of Practice. Translated by Nice, R. New York, London, Melbourne: Cambridge University Press.

— (1989). 'Social Space and Symbolic Power.' Sociological Theory 7, no. 1: pp.14-25.

Branaman, A. (1997). 'Goffman's Social Theory.' In The Goffman Reader, edited by Branaman, A. and C. Lemert, pp.xivi-lxxxii. Malden VA; Oxford: Blackwell Publishing.

Braudel, F. (1982). Civilization & Capitalism 15th-18th Century: Perspectives of the World. Vol. 3. London: Phoenix Press.

Brooks, F. (1995). The Mythical Man-Month: Essays on software engineering (20th anniversary edition). Reading, MA: Addison-Wesley.

Brooks, F. P. J. (1975). The Mythical Man Month. Reading, MA: Addison-Wesley.

Calhoun, C. (1980). 'Community: Toward a variable conceptualization for comparative research.' Social History 5: pp.105-129.

Carrier, J. G. (1995). Gifts and Commodities: Exchange and Western Capitalism since 1700. New York: Routledge.

Carter, B. T. (2001). 'The Complexity of Social Networks: Theoretical and empirical findings.' Social Networks 23: pp.31-71.

Castells, M. (2000a). 'Materials for an explanatory theory of the Network Society.' British Journal of Sociology 51, no. 1: pp.5-24.

— (2000b). The Information Age: Economy, Society and Culture Volume 1: The rise of the network society. Oxford: Blackwell.

— (2001). The Internet Galaxy. Oxford; New York: Oxford University Press.

Cheal, D. (1988). The Gift Economy. London and New York: Routledge.

Christensen, S. and A. Westenholtz (1999). 'Boards of Directors as Strategists in an Enacted World-The Danish Case.' Journal of Management and Governance 3, no. 3: pp.261-268.

Clegg, R. S. (1989). Frameworks of Power. London;A Thousand Oaks, CA: Sage Publications.

Coase, R. (1937). 'The Nature of the Firm.' Economica 4, no. 16: pp.pp. 386-405.

Coleman, G. (2004). 'The Political Agnosticism of Free and Open Source Software and the Inadvertent Politics of Contrast.' Anthropological Quarterly 77, no. 3: pp.507-519.

— (2005). 'Three Ethical Moments in Debian: the Making of the Ethical Hacker.' Chapter 6 in 'The Social Construction of Freedom in Free and Open Source Software: Hackers, Ethics and the Liberal Tradition'. Doctoral Dissertation, Department of Anthropology, University of Chicago.

Contu, A. and H. Willmott (2003). 'Re-embedding Situatedness: The Importance of Power Relations in Learning Theory.' Organization Science 14, no. 3: pp.283-296.

Cooper, M. N. (March 2005). 'Time for the Recording Industry to Face the Music: The political, social and economic benefits of peer-to-peer communications networks'. Stanford Law School Centre for Internet and Society.

Cross, R. and A. Parker (2004). The Hidden Power of Social Networks: Understanding how work really gets done in organizations. Boston, MA: Harvard Business School Press.

Crowston, K. and J. Howison (2005). 'The Social Structure of Free and Open Source Software Development.' First Monday: http://www.firstmonday.org/issues/issue10_2/crowston/ , 10, no2, last accessed 05/04/07.

Currah, A. (2006). 'Hollywood versus the Internet: The media and entertainment industries in a digital and networked economy.' Journal of Economic Geography 6: pp.439-468.

Dafermos, G. N. (2001). 'Management and Virtual Decentralised Networks: The Linux Project.' First Monday: http://www.firstmonday.dk/issues/issue6_11/dafermos/index.html#author, 6 no 11, last accessed 26/06/04.

Dahlander, L. (2004). 'Appropriating the Commons: Firms in Open Source Software.' Online paper available at http://opensource.mit.edu/papers/dahlander2.pdf, last accessed 22/02/07.

Dahlander, L. and G. M. Magnusson (2005). 'Relationships Between Open Source Software Companies and Communities: Observations from Nordic Firms.' Research Policy, no. 34: pp.481-493.

Dalle, J.-M. and P. David (2005). 'The Allocation of Software Development Resources in Open Source Production Mode.' In Perspectives on Free and Open Source Software, edited by Feller, J., B. Fitzgerald, S. Hissam and K. R. Lakhani, pp.297-238. Cambridge, MA; London, UK: The MIT Press.

Dalle, J.-M., P. David, R. A. Ghosh and E. W. Steinmueller (2005). 'Advancing Economic Research on the Free and Open Source Model of Production.' In How Open is the Future? Economic, Social and Cultural Scenarios inspired by Free & Open Source Software, edited by Wynants, M. and J. Cornelis, pp.395-426. Brussels: VUB-Brussels University Press.

David, P., A. Waterman and S. Arora. (2003). FLOSS-US/ The Free/Libre/Open Source Developer Software Survey for 2003: A First Report. http://www.stanford.edu/group/floss-us/report/FLOSS-US-Report.pdf. last accessed 5/5/04.

De Landa, M. (2000). 1,000 Years of Non-Linear History. New York: Swerve Editions.

Demil, B. and X. Lecocq (2006). 'Neither Market, nor Hierarchy or Network: The emerging bazaar governance.' Organization Studies 27, no. 10: pp.1447-1466.

Divitini, M., L. Jaccheri, E. Monteiro and H. Traetteberg (2003). 'Open source Processes: No place for politics?' 3rd Workshop on Open Source Software Engineering-ICSE'03 International Conference on Software Engineering

http://opensource.ucc.ie/icse2003/3rd-WS-on-OSS-Engineering.pdf ,last accessed 5/5/04: pp.39-44.

Dravis, P. (2003). 'Open Source Software: Perspectives for Development-Report presented in the Information for Development Symposium.' 9-10 December 2003, Geneva.

Ducheneaut, N. (2005). 'Socialization in an Open Source Software Community: a Socio-technical Analysis.' Computer Supported Cooperative Work 14, no. 4: pp.323-368.

Edquist, C., ed. (1997). Systems of Innovation: Technologies, Institutions, and Organizations. London: Pinter.

Elliot, M. S. and W. Scacchi (2003a). 'Free Software Developers as an Occupational Community: Resolving Conflicts and Fostering Collaboration.' in revised version in ACM Group'03 Conference Proceedings, Sanibel Island, FL 9-12, (eds),

---. (2003b). Free Software: a Case Study of Sofware Development in a Virtual Organization Culture. Working Paper, Institute for Software Research, UC Irvine, April 2003, available at: http://www.ics.uci.edu/%7Ewscacchi/Papers/New/Elliott-Scacchi-GNUe-Study-Report.pdf (accessed 6/06/04).

Feller, J. and B. Fitzgerald (2002). Understanding open source software development. London; Boston; New York: Addison-Wesley.

Field, A. (2005). Discovering Statistics using SPSS. 2nd Edition ed. London: Sage Publications Ltd.

Fitzgerald, B. (2006). 'The Transformation of Open Source Software.' MIS Quarterly 30, no. 3: pp.587-598.

Flick, U. (2002). An Introduction to Qualitative Analysis. London, Thousand Oaks, New Delhi: Sage Publications.

Foucault, M. (1979). Discipline and Punish: The Birth of the Prison. Harmondsworth: Penguin.

--- (1977). Language, Counter-Memory, Practice: Selected Essays and Interviews. Oxford: Blackwell.

— (2002a). 'Questions of Method.' In Power/Michel Foucault, edited by Faubion, J. D., pp.223-238. New York: New Press.

--- (2002b). 'Space, Knowledge and Power.' In Michel Foucault: Power/ Essential works of Foucault 1954-1984. Vol.1 edited by Faubion, J. D., pp.349-364. London; New York: New Penguin Books.

--- (1982a). 'Technologies of the Self.' In Michel Foucault: Ethics/ Essential works of Foucault 1954-1984. Vol.1, edited by Faubion, J. D., pp.223-251. London; New York: Penguin Books.

— (1981). 'The Order of Discourse.' In Untying the text: a post-structuralist reader, edited by Young, R., pp.48-78. Boston; London: Routledge and Kegan Paul.

--- (1982b). 'The Political Technology of Individuals.' In Michel Foucault: Power/ Essential works of Foucault 1954-1984. Vol. 3, edited by Faubion, J. D., pp.403-417. London; New York: Penguin Books.

--- (1982c). 'The Subject and Power.' In Michel Foucault: Power/ Essential works of Foucault 1954-1984. Vol. 3, edited by Faubion, J. D., pp.326-348. London; New York: Penguin Books.

--- (1980). 'Two Lectures.' In Power/Knowledge: Selected Interviews and Other Writings, 1972-1977, edited by Gordon, C., pp.78-108. Brighton: Harvester Press.

Foucault, M. and C. Gordon (1980). Power-Knowledge: Selected Interviews and Other Writings, 1972-1977. Brighton: Harvester Press.

Foucault, M. and P. H. Hutton (1988). Technologies of the Self: A Seminar with Michel Foucault. London: Tavistock.

Fox, S. (2000). 'Communities of Practice, Foucault and Actor Network Theory.' Journal of Management Studies 37, no. 6: pp.853-867.

Franck, E. and C. Jungwirth (2003). 'Reconciling Investors and Donators: the governace structure of open source projects.' Journal of Management and Governance 7: pp.401-421.

Freeman, C. (1982). The economics of industrial innovation. London: Pinter.

Freeman, C. and L. Soete, eds (1987). Technical Change and Full Employment. Oxford; New York: Basil Blackwell Ltd.

Freeman, S. and J. Siltala. (2004). Freedom and Profit: How suits and hackers are working it out on the desktop. Working paper presented in 4/EASST Joint Meeting, Paris 26/08/04 http://personal.inet.fi/koti/jsiltala/juha/floss/freedom-profit-paper.pdf last accessed 19/01/06.

Frey, B. S. and F. Jegen (2001). 'Motivation Crowding Theory.' Journal of Economic Surveys 15, no. 5.

Friedman, N. and M. Icaza (2005). 'Mechanics of Open Source: Growing and Harvesting Your Project ' Talk to the Massachussets Software Council, June 2005 available at http://www.nat.org/2005/june/MassSoftware.pdf, last accessed 22/09/06.

Fuggetta, A. (2003). 'Open Source Software-An Evaluation.' The Journal of Systems and Software, no. 66: pp.77-90.

Garcia, M. J. and E. W. Steinmueller (2003a). 'Applying the Open Source Model to Knowledge Work.' INK Open Source Research Working Paper No. 2, SPRU-Science and Technology Policy Research, University of Sussex, http://siepr.stanford.edu/programs/OpenSoftware_David/OSR02_abstract.html last accessed 14/06/04.

— (2003b). 'The Open Source Way of Working: A New Paradigm for the Division of Labour in Software Development?' INK Open Source Research Working Paper No.92, SPRU-Science and Technology Policy Research, University of Sussex available at http://siepr.stanford.edu/programs/OpenSoftware_David/NSFOSF_Publications.html.

Geertz, C. (1979). The Interpretation of Cultures. New York: Basic Books.

Geertz, C., H. Geertz and L. Rosen (1979). Meaning and Order in Moroccan Society: Three Essays in Cultural Analysis. Cambridge: Cambridge University Press.

German, D. M. (2002). 'The Evolution of the Gnome Project.' Workshop Proceeding presented in 'Meeting Challenges and Surviving Success: the 2nd Workshop on Open Source Software Engineering' (May 19-25) http://opensource.ucc.ie/icse2002/German.pdf last accessed 5/5/04.

Ghosh, R. A. and P. A. David (2003). 'The Nature and Composition of the Linux Kernel Developer Community: A dynamic analysis.' SIEPR-Project NOSTRA Working Paper, draft version 5, http://dxm.org/papers/licks1/licksresults.pdf
last accessed 12/06/04.

Ghosh, R. A., R. Glott, B. Krieger and G. Robles (2002). 'FLOSS: Free/Libre and Open Source Software Survey and Study: Part 4: Survey of Developers'. International Institute of Infonomics, University of Maastricht.

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

Goffman, E. (1974). Frame Analysis. Middlesex: Penguin Books.

— (1969a). 'On Face Work.' In Where the Action is-Three Essays, pp. 4-36. London: Allen Lane-The Penguin Press.

— (1970). Strategic Interaction. Oxford: Blackwell.

— (1969b). Where the Action is-Three Essays. London: Allen Lane-The Penguin Press.

Goody, E. (1989). 'Learning, Apprenticeship and the Division of Labor.' In Apprenticeship: From Theory to Method and Back Again, edited by Coy, M. W., pp.233-256. Albany NY: State University of New York Press.

Granovetter, M. (1985). 'Economic Action and Social Structure: The Problem of Embeddedness.' The American Journal of Sociology 91, no. 3: pp.481-510.

Granovetter, M. and R. Swedberg, eds (2001). The Sociology of Economic Life-2nd ed. Boulder: Westview Press.

Hacking, I. (2004). 'Between Michel Foucault and Erving Goffman: Between Discourse in the Abstract and Face-to-Face Interaction.' Economy and Society 33, no. 3: pp.277-302.

Handy, C. (1995). 'Trust and the Virtual Organization.' Harvard Business Review 73, no. 3: pp.40-50.

Hann, I.-H., J. Roberts, S. Slaughter and R. T. Fielding (2004). 'An Empirical Analysis of Economic Returns to Open Source Participation.' Unpublished Working Paper, Carnegie-Mellon University, available at: http://www.andrew.cmu.edu/user/jroberts/Paper1.pdf , last accessed 12/2006.

Haruvy, E., A. Prasad and S. P. Sethi (2003). 'Harvesting Altruism in Open-Source Software Development.' Journal of Optimization Theory and Applications 118, no. 2: pp.381-416.

Hass, P. M. (1992). 'Introduction: Epistemic Communities and International Policy Coordination.' International Organization 46, no. 1: pp.1-35.

Hemetsberger, A. (2004). 'Fostering Cooperation on the Internet: Social Exchange Processes in Innovative Virtual Consumer Communities.' Paper presented at the annual ACR (Association for Consumer Research) conference 2001 in Austin, Texas, also available online at: http://opensource.mit.edu/papers/hemetsberger2.pdf.

Hemetsberger, A. and C. Reindhart (2004). Sharing and Creating Knowledge in Open Source Communities: the case of KDE, Abstract accepted for the 5th Conference on Organizational Knowledge, Learning and Capabilities in Innsbruck, Austria, 2004.

Henriques, J., W. Holloway, C. Urwin, C. Venn and V. Walkerdine (1984). Changing the Subject: Psychology, Social Regulation and Subjectivity. London: Methuen.

Herzenberg, S. (1998). New rules for a New Economy: Employment and Opportunity in Postindustrial America. Ithaca: ILR Press.

Himanen, P. (2001). The Hacker Ethic and the Spirit of the Information Age. London: Vintage.

Hook, D. (2001). 'Discourse, Knowledge, Materiality, History-Foucault and Discourse Analysis.' Theory and Psychology 11, no. 4: pp.521-547.

Howison, J., I. Keisuke and K. Crowston (2006). 'Social Dynamics of Free and Open Source Team Communication.' in Open Source Systems-IFIP Working Group 2.13, Foundation on Open Source Software Proceedings, Como, Italy, Damiani, E., B. Fitzgerald, W. Scacchi, M. Scotto and G. Succi (eds), Springer.

Huang, S.-K. and K.-m. Liu (2005). 'Mining Version Histories to Verify the Learning Process of Legitimate Peripheral Participants.' in MSR 2005: 2nd International Workshop on Mining Software Repositories Proceedings, Saint Louis, Missouri, USA, 17th May pp.84-89.

Huhtala, H. M. J. (2004). 'The Emancipated Worker? A Foucauldian Study of Power, Subjectivity and Organizing in the Information Age.' Doctoral Dissertation The Finnish Science Society. Commentationes Scientiarum Socialicum Series.

European Union (2006). Lisbon Declaration: an alliance for social cohesion through digital inclusion. Lisbon: 28-29 April.

Jarvernpaa, S. L., K. Knoll and D. Leidner (1998). 'Is Anybody Out There?: Antecendents of Trust in Global Virtual Teams.' Journal of Management Information Systems 14, no. 4: pp.29-64.

Kasper-Fuehrera, E. C. (2001). 'Communicating Trustworthiness and Building Trust in Interorganizational Virtual Organizations.' Journal of Management 27, no. 3: pp.235-254.

Kendall, G. and G. Wickham (1998). Using Foucault's Methods. London: Sage.

Kidder, T. (2000). The Soul of a New Machine. Boston: Back Bay Books.

Knorr-Cetina, K. (1999). Epistemic Cultures. Cambridge, MA: Harvard University Press.

Koch, S. and G. Schneider (2002). 'Effort, Co-ordination and Co-operation in an Open Source Software Project: GNOME.' Information Systems Journal 12, no. 1: pp.27-42.

Kollock, P. (1999). 'The Economies of Online Cooperation: Gifts and Publics Goods in Cyberspace.' In Communities in Cyberspace, edited by Smith, M. and P. Kollock, pp.220-242. London: Routledge.

Krishmamurti, S. (2005). 'An Analysis of Open Source Business Models.' In Perspectives on Free and Open Source Software, edited by Feller, J., B. Fitzgerald, S. A. Hissam and K. R. Lakhani, pp.279-296. Cambridge MA: The MIT Press.

— (2002). 'Cave or Community? An Empirical Examination of 100 Mature Open Source Projects.' First Monday http://www.firstmonday.dk/issues/issue7_6/krishnamurthy/index.html last accessed 5/5/04 6, no. 7.

Krishmamurti, S. and A. Tripathi (2005). 'Bounty Programs in Free/Libre/Open source Software (FLOSS): An Economic Analysis".' In The Economics of Open Source Software Development (Forthcoming), edited by Bitzer, J. and P. Schroeder: Elsevier Publications.

Kunda, G., S. Barley R. and J. Evans (2002). 'Why Do Contractors Contract? The Experience of Highly Skilled Technical Professionals in a Contigent Labour Market.' Industrial and Labor Relations Review 55, no. 2: pp.234-261.

Lakhani, K. (2006). 'The Core and Periphery in Distributed and Self-organizing Systems.' Doctoral Dissertation, Sloan School of Management, MIT.

Lakhani, K. and B. Wolf. (2002). The Boston Consulting Group/OSDN Hacker Survey. OSDN-Open Source Development network, http://www.osdn.com/bcg last accessed 3/04/04.

Lakhani, K. and R. G. Wolf (2005). 'Why hackers do what they do: Understanding Motivation Effort in Free/Open Source Software Projects.' In Perspectives on Free and Open Source Software, edited by Feller, J., B. Fitzgerald, S. Hissam and K. R. Lakhani, pp.3-22. Cambridge, MA; London, UK: The MIT Press.

Lancashire, D. (2001). 'Code, Culture and Cash: The Fading Altruism of Open Source Development.' First Monday, peer-reviewed journal on the Internet http://firstmonday.org/issues/issue6_12/lancashire/index.html, 6, no. 12, last accessed 22/02/2007.

Lansing, J. S. (2002). 'Artificial Societies and the Social Sciences.' Santa Fe Institute Working Paper 02-03-011.

Lee, G. and R. Cole (2003). 'From a Firm Based to a Community Based Model of Knowledge Creation: The case of The Linux Kernel.' Organization Science 14: pp.633-649.

Lerner, J. and J. Tirole (2002a). 'Some Simple Economics of Open Source.' Journal of Industrial Economics 50, no. 2: pp. 197-234.

— (2005). 'The Economics of Technology Sharing: Open Source and Beyond.' Journal of Economic Perspectives 19, no. 2: pp.99-120(122).

— (2002b). 'The Scope of Open Source Licensing.' NBER Working Paper No. W9363. http://ssrn.com/abstract=359303.

Lesser, E. L., M. A. Fontainer and J. A. Slusher, eds (2000). Knowledge and Communities. Boston MA: Butterworth-Heineman.

Lessig, L. (2002). 'Open Source Baselines: Compared to What?' In Government Policy toward Open Source Software, edited by Hahn, R. W., pp.50-68. Washington D.C.: AEI-Brookings Joint Centre For Regulatory Studies.

Levy, S. (1984). Hackers: Heroes of the Computer Revolution. Garden City, NY: Doubleday.

Leyshon, A. (2003). 'Scary Monsters? Software Formats, Peer-to-Peer Networks and the Spectre of the Gift.' Environment and Planning D: Society and Space 21: pp.533-558.

Lin, Y. (2005). 'Hybrid Innovation: How Does the Collaboration Between the FLOSS Community and Corporations Happen? XVIV no. 1. Forthcoming in Knowledge, Technology and Policy Summer 2005

Ljundgberg, J. (2000). 'Open Source Movements as a model of organizing.' European Journal of Information Systems 9, no. 4: pp.208-216.

Lopez-Fernandez, L., G. Robles and J. M. Gonzalez-Borahona (2004). 'Applying Social Network Analysis to the Information in CVS repositories.' in MSR 2004: 1st International Workshop on Mining Software Repositories Proceedings, Edinburgh, Scotland, UK, 24th May 2004, Hassan, A. E., R. C. Holt and A. Mockus (eds), pp.101-105.

Lundvall, B.-A. (2002). Innovation, Growth and Social Cohesion: the Danish Model. Cheltenham: Edward Elgar.

Madanmohan, T. R. and S. Navelkar (2002). 'Roles and Knowledge Management in Online Technology Communities: An Ethnographic Study.' Online paper available at http://opensource.mit.edu/papers/madanmohan2.pdf, last accessed 23/07/2003.

Mauss, M. (1954). The Gift: The Form and Reason for Exchange in Archaic Societies. London and New York: Routledge Classics.

McKinlay, A. (2002). 'The Limits of Knowledge Management.' New Technology, Work and Employment 17, no. 2: pp.76-87.

McLennan, G. (2001). 'Sociology's Complexity.' Sociology 37, no. 3: pp.547-564.

McNamee, S. J. and R. J. Miller (2004). The Meritocracy Myth. Lanham, Md: Rowman & Littlefield Publishers, Inc.

Meyrowitz, J. (1985). No Sense of Place: The Impact of Electronic Media on Social Behaviour. New York: Oxford University Press.

Michlmayr, M. and M. B. Hill (2003). 'Quality and Reliance on Individuals in Open Source Projects.' in 3rd Workshop on Open Source Software Engineering-ICSE'03 International Conference on Software Engineering. Available at:

http://opensource.ucc.ie/icse2003/3rd-WS-on-OSS-Engineering.pdf, last accessed 5/5/04 Proceedings, Portland, Oregon, USA, Feller, J., B. Fitzerald, S. Hissam and K. Lakhani (eds), pp.105-110.

Miller, H. (1995). 'The presentation of Self in Electronic Life: Goffman on the Internet.' Paper presented at Embodied Knowledge and Virtual Space Conference, Goldsmith's College, University of London, June 1995.

Miller, H. and J. Arnold (2001). 'Breaking away from Grounded Identity? Women Academics on the Web.' CyberPsychology & Behaviour 4, no. 1: pp.95-108.

Mockus, A., R. T. Fielding and Herbsleb (2002). 'Two Case Studies of Open Source Software Development: Apache and Mozilla.' ACM Transactions on Software Engineering and Methodology 11, no. 3: pp.pp.309-346.

Moon, Y. J. and L. Sproull (2002). 'Essence of Distributed Work: The Case of the Linux Kernel.' In Distributed Work, edited by Hinds, P. and S. Kiesler, pp.381-404. Cambridge, MA: MIT Press.

Munro, I. (2000). 'Non Disciplinary Power and the Network Society.' Organization 7, no. 4: pp.679-695.

Musamichi, T., M. Fujimoto and N. Yamasaki (2003). 'The Active Lurker: a New Viewpoint for Evaluating the Influence of an In-house Online Community.' Working Paper for the GROUP2003 Workshop on 'Communities in Context'.

Mustonen, M. (2005). 'When Does a Firm Substitute Open Source Programming?' Journal of Economics and Management Strategy 14, no. 1: pp.121-139.

Neamtiu, I., J. S. Foster and M. Hicks (2005). 'Understanding Source Code Evolution Using Abstract Syntax Tree Maching.' in MSR 2005: 2nd International Workshop on Mining Data Proceedings, Saint Louis, Missouri, USA, Hassan, A. E., R. C. Holt and S. Diehl (eds), pp.2-6.

Newmarch, J. (2001). 'Lessons from Open Source: Intellectual Property and Software.' FirstMonday: http://www.firstmonday.dk/issues/issue6_6/newmarch/index.html, no. 6, 6, last accessed 26/06/04.

O'Mahony, C. S. (2003). 'Guarding the Commons: How Community Managed Software Projects Protect their Work.' Research Policy 32: pp.1168-1198.

— (2002). 'The Emergence of a New Commercial Actor: Community Managed Software Projects.' Doctoral Dissertation, Department of Management Science and Engineering Management, Stanford University.

O'Mahony, C. S. and F. Ferraro (2003). 'Managing the Boundary of an 'Open' Project.' Draft 2.0 prepared for the Santa Fe Institute Workshop (SFI) on the Network Construction of Markets.

O'Mahony, C. S. and J. West (2005). 'Contrasting Community Building in Sponsored and Community Founded Open Source Projects.' Proceedings of the 38th Annual Hawai'i International Conference on System Sciences, Waikoloa, Hawaii, January 3-6, 2005.

Offer, A. (1997). 'Between the Gift and the Market: The Economy of Regard.' Economic History Review 3: pp.450-476.

Orr, J. E. (1996). Talking about Machines: An Ethnography of a Modern Job. Ithaca, N.Y.: ILR Press.

Pavitt, K. (1999). Technology, Management and Systems of Innovation. Cheltenham: Edward Elgar.

Polanyi, K. (1944). The Great Transformation: the Political and Economic Origins of Our Time. Boston: Beacon Press.

Poster, M. (1990). 'Foucault and Databases: Participatory Surveillance.' In The Modes of Information: Poststructuralism and Social Context, edited by Poster, M., pp.69-98. Chicago IL: University of Chicago Press.

— (1984). Foucault, Marxism and history: Mode of Production Versus Mode of Information, Social and political theory. Cambridge: Polity.

Potter, J. and M. Wetherell (1987). Discourse and Social Psychology: Beyond Attitudes and Behaviour. London: Sage.

Preece, J., B. Nonnecke and D. Andrews (2004). 'The top five reasons for lurking: improving community experiences for everyone.' Computers in Human Behavior 20, no. 2: pp.201-233.

Quintas, P., ed. (1993). Social Dimensions of Systems Enginnering: People, Processes and Software Development. Interactive Information Systems. New York; London; Toronto; Sydney; Tokyo; Singapore: Ellis Horwood.

Raymond, E. (2000a). 'Homesteading the Noosphere.' version 3.0 available at http://www.catb.org/-esr/writings/homesteading/homesteading.

---. (2000b). The Cathedral and the Bazaar. http://tuxedo.org/-esr/writings/cathedral-bazaar/, last acccessed 22/02/2003.

Raymond, E. S. (2001a). The Cathedral and the Bazaar. Edited by Raymond, E. S., The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary. Sebastopol, CA: O'Reilly.

— (2001b). 'The Magic Cauldron.' In The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary, edited by Raymond, E. S., pp.pp. 113-166. Sebastopol, CA: O' Reilly.

Rheingold, H. (1993). The Virtual Community: Homesteading on the Electronic Frontier. Reading, MA: Addison-Wesley.

Roberts, J., I.-H. Hann and S. Slaughter (2006a). 'Communication Networks in an Open Source Software Project.' Paper presented at the The Second International Conference on Open Source Systems, 8-10 June 2006, Como, Italy.

— (2006b). 'Understanding the Motivations, Participation, and Performance of Open Source Software Developers: a Longitudinal Study of the Apache Projects.' Management Science 52, no. 7: pp.984-999.

Rose, N. (1999). Powers of Freedom: Reframing Political Thought. Cambridge: Cambridge University Press.

Samer, F. and M. Wasko McLure (2002). 'The Web of knowledge: An investigation of Knowledge Exchange in Networks of Practice.' Online paper available at http://opensource.mit.edu/papers/Farajwasko.pdf, last accessed on 23/07/2003.

Sassen, S. (2004). 'Cited Materialities with Global Span.' In Society Online: the Internet in Context, edited by Howard, P. N. and S. Jones, pp.295-306. London: Sage.

--- (2005a). 'Electronic Markets and Activist Networks: The Weight of Social Logics in Digital Formations.' In Digital Formations: IT and New Architectures in the Global Realm, edited by Sassen, S. and R. Latham, pp.54-88. Princeton NJ: Princeton University Press.

--- (2005b). 'The Embeddedness of Electronic Markets: The Case of Global Capital Markets.' In The Sociology of Financial Markets, edited by Knorr-Cetina, C. and A. Preda, pp.38-61. Oxford; New York: Oxford University Press.

--- (2002). 'Towards a Sociology of Information Technology.' Current Sociology 50, no. 3: pp.365-388.

Schaefer, R. (2006). 'A Critical Programmer Searches for Professionalism.' ACM SIGSOFT Software Engineering Notes 31, no. 4.

Schmidt, K. M. and M. Schnitzer (2002). 'Public Subsidies for Open Source? Some Economic Policy Issues for the Software Market.' Available at http://papers.ssrn.com/sol3/papers.cfm?abstract_id=319908 last accessed 01/05/2004.

Sewell, G. (1998). 'The Discipline of Teams: The Control of Team Based Industrial Work through Electronic and Peer Surveillance.' Administrative Science Quarterly 43: pp.397-428.

Shah, S. (2003). 'Understanding the Nature of Participation & Coordination in Open and Gated Source Software Development Communities.' Dissertation Chapter, University of Illinois, Urbana-Champaign. Available at: http://opensource.mit.edu/papers/shah3.pdf, last accessed 10/-7/04.

Shaikh, M. and T. Cornford (2003). 'Version Control Management Tools: CVS to BK in the Linux Kernel.' 25th International Conference on Software Engineering proceedings - Taking Stock of the Bazaar: The 3rd Workshop on Open Source Software Engineering, Portland, Oregon, May 3rd: pp. 127-132.

— (2004). 'Version Control Tools: a Collaborative vehicle for learning in F/OS.' 6th International Conference on Software Engineering proceedings - Collaboration, Conflict and Control: The 4th Workshop on Open Source Software Engineering, Edinburgh, May 25th.

Shapiro, C. and H. R. Varian (1999). Information Rules: A Strategic Guide to the Network Economy. Boston, MA: Harvard Business School Press.

Sharma, S., V. Suguraman and B. Rajagopalani (2002). 'A Framework for Creating Hybrid Open Source Communities.' Information Systems Journal 12: pp.7-25.

Silverstone, R. (2005). 'The Sociology of Mediation and Communication.' In The Sage Handbook of Sociology, edited by Calhoun, C., C. Rojek and B. Turner, pp.188-207. London; A Thousand Oaks; New Delhi: Sage Publications.

Simpson, I. H. (1999). 'Historical Patterns of Workplace Organization: from Mechanical to Electronic Control and Beyond.' Current Sociology 47, no. 2: pp.47-75.

Slater, D. (2002). 'Social Relationships and Identity Online and Offline.' In The Handbook of New Media, edited by Lievrouw, L. and S. Livingstone, pp.534-547. London: Sage.

Sliweski, J., T. Zimmerman and A. Zeller (2005). 'When do Changes Induce Fixes.' in MSR 2005: 2nd International Workshop on Mining Software Repositories Proceedings, Saint Louis, Missouri, USA, Hassan, A. E., R. C. Holt and S. Diehl (eds), pp.24-28.

Smith, G., ed. (1999). Goffman and Social Organization: Studies in a Sociological Legacy. London and New York: Routledge.

Spaeth, S. (2005). 'Coordination in Open Source Projects: a Social Network Analysis using CVS data.' Doctoral Dissertation, Graduate School of Business Administration, Economics, Law and Social Sciences, University of St. Gallen.

Stalder, F. and J. Hirsh (2002). 'Open Source Intelligence.' First Monday: http://www.firstmonday.dk/issues/issue7_6/index.html, no 6, last accessed 7/06/04.

Stewart, D. (2004). 'Status Inertia: The Speed Imperative in the Attainment of Community Status.' Online paper available at: opensource.mit.edu/papers/stewart2.pdf, last accessed on 2/06/04.

Stewart, K. J. and S. Gosain (2006). 'The Impact of Ideology, Trust and Communication on Effectiveness in Open Source Software Development Teams.' MIS Quarterly 30, no. 2: pp.291-314.

Studenmund (2001). Using econometrics: a practical guide. Boston, MA: Addison-Wesley.

Swan, J., H. Scarbrough and M. Robertson (2002). 'The Construction of 'Communities of Practice' in the Management of Innovation.' Management Learning 33, no. 4: pp.477-496.

Swartz, D. (1997). Culture & Power : the Sociology of Pierre Bourdieu. Chicago: University of Chicago Press.

Thompson, J. B. (1995). The Media and Modernity: a Social Theory of the Media. Cambridge: Polity Press.

— (1989). 'The Theory of Structuration.' In Social Theory of Modern Societies- Anthony Giddens and his critics, edited by Held, D. and J. B. Thompson, pp.56-76. Newcastle upon Twine: Cambridge University Press.

Thrift, N. (2005). Knowing Capitalism. London, UK; a Thousand Oaks, CA: Sage Publications.

Titmuss, R. (1997). The Gift Relationship: From Human Blood to Social Policy. Edited by Oakley, A. and J. Ashton. London: London School of Economics and Political Science.

Turkle, S. (1996). Life on the Screen: Identity in the Age of the Internet. London: Weidenfeld & Nicolson.

Turnbull, D. (2000). Masons, Tricksters and Chartographers. Brighton: Harwood Academic Publishers.

United Nations ICT Task Force (2004). The Role of Information and Communication Technology in Global Development-Analyses and Recommendations, ICT task force series 3. New York: United Nations: UNICT.

--- (2005). The World Summit on the Information Society: Moving from the Past into the Future. Edited by Stauffacher, D. and W. Kleinwachter, United Nations ICT Task Force, series 8. New York: United Nations: UNICT.

Uzzi, B. (2001). 'Social Structure and Competition in Interfirm Networks: The Paradox of Embeddedness ' In The Sociology of Economic Life-2nd ed., edited by Granovetter, M. and R. Sweberg, pp.207-238. Boulder, CO: Westview Press.

--- (1996). 'The Sources and Consequences of Economic Embeddedness in Organizations: the Network Effect.' American Sociological Review 6, no. 4: pp.674-698.

Vallas, S. P. (1999). 'Rethinking Post-Fordism: The Meaning of Workplace Flexibility.' Sociological Theory 17, no. 1: pp.68-101.

Veale, J. K. (2005). 'Internet Gift Economies: Voluntary Payment Schemes as Tangible Reciprocity.' First Monday: http://www.firstmonday.org/issues/issue8_12/veale/, last accessed 02/03/2006 8, no. 12.

Villa, L. (2003). 'GNOME Organization and Teams.' Speech given on 17/06 at GUADEC (GNOME Users and Developers Conference) 2003, Dublin, Ireland.

von Hippel, E. (2002). Horizontal Innovation Networks-by and for users. MIT Sloan School of Management, Working Paper No. 4366-02, Boston, MA (accessed 23/07/2003).

--- (2005). 'Open Source Software Projects as "User Innovation Networks".' In Perspectives on Free and Open Source Software, edited by Feller, J., B. Fitzgerald, S. Hissam and K. R. Lakhani, pp.267-278. Cambridge, MA: The MIT Press.

von Hippel, E. and G. von Krogh (2003). 'Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science.' Organization Science 14, no. 2: pp.209-223.

von Krogh, G., S. Spaeth and K. R. Lakhani (2003). 'Community, Joining and Specialization in Open Source Software Innovation: a Case Study.' Research Policy 32: pp.1217-1241.

Walker, K. (2000). '"Its Difficult to Hide It": The Presentation of Self on Internet Home Pages.' Qualitative Sociology 23, no. 1: pp.99-120.

Weber, S. (2000). The Political Economy of Open Source. Working Paper Series 1011, UCAIS Berkeley Roundtable on the International Economy, UC Berkeley http://repositories.cdlib.org/cgi/viewcontent.cgi?article=1011&context=brie, last accessed 23/09/05.

— (2004). The Success of Open Source. Cambridge, MA: Harvard University Press.

Webster, F. (1995). Theories of the Information Society. London and New York: Routledge.

Weerawarana, S. and J. Weeratunga (2004). Open Source in Developing Countries. Stockholm: Swedish International Development Cooperation Agency (SIDA).

Wellman, B. (2004). 'The Three Ages of Internet Studies: Ten, Five and Zero Years Ago.' New Media & Society 6, no. 1: pp.123-129.

Wenger, E. (1998). Communities of Practice: Learning, Meaning and Identity. Cambridge: Cambridge University Press.

Wenger, E. and J. Lave (1991). Situated Learning: Legitimate Peripheral Participation. Cambridge: Cambridge University Press.

Wenger, E., R. McDermot and W. Snyder M. (2002). Cultivating Communities of Practice: A Guide to Managing Knowledge. Boston MA: Harvard Business School Press.

West, J. (2003). 'How Open is Open Enough? Melding Proprietary and Open Source Platform Strategies.' Research Policy 32, no. 7: pp.1259-1285.

West, J. and C. S. O'Mahony. (2004). Contrasting Community Building in Sponsored and Community Founded Open Source Projects. http://opensource.mit.edu/papers/westomahony.pdf, last accessed 25/07/04.

White, R. and D. Hanson (2001). 'Corporate Self, Corporate Reputation and Corporate Annual Reports: Re-enrolling Goffman.' Scandinavian Journal of Management 18, no. 3: pp.285-301.

Wickham, G., G. C. Pavlich and Onati International Institute for the Sociology of Law. (2001). Rethinking Law, Society and Governance: Foucault's Bequest, Oñati international series in law and society. Oxford: Hart.

Willcocks, L. P. (2006). 'Michel Foucault and the Social Study of ICTs: Critique and Reappraisal.' Department of Information Systems, London School of Economics and Political Science Working paper series no 138 available at: http://is2.lse.ac.uk/wp/pdf/WP138.PDF, last accessed October 2006.

Williamson, E. O. (1987). The Economic Institutions of Capitalism: Firms, Market, Relational Contracting. New York: The Free Press.

— (2000). 'The New Institutional Economics: Taking Stock, Looking Ahead.' Journal of Economic Literature 38, no. September: pp.595-613.

— (1979). 'Transaction Cost Economics: The Governance of Contractual Relations.' Journal of Law and Economics 22, no. 2: pp.233-261.

Wilson, F. (1995). 'Managerial Control Strategies within the Network Organization.' Information, Technology & People 8, no. 3: pp.57-72.

Yin, R. K. (2003). Case Study Research: Design and Methods. A Thousand Oaks, CA: Sage Publications.

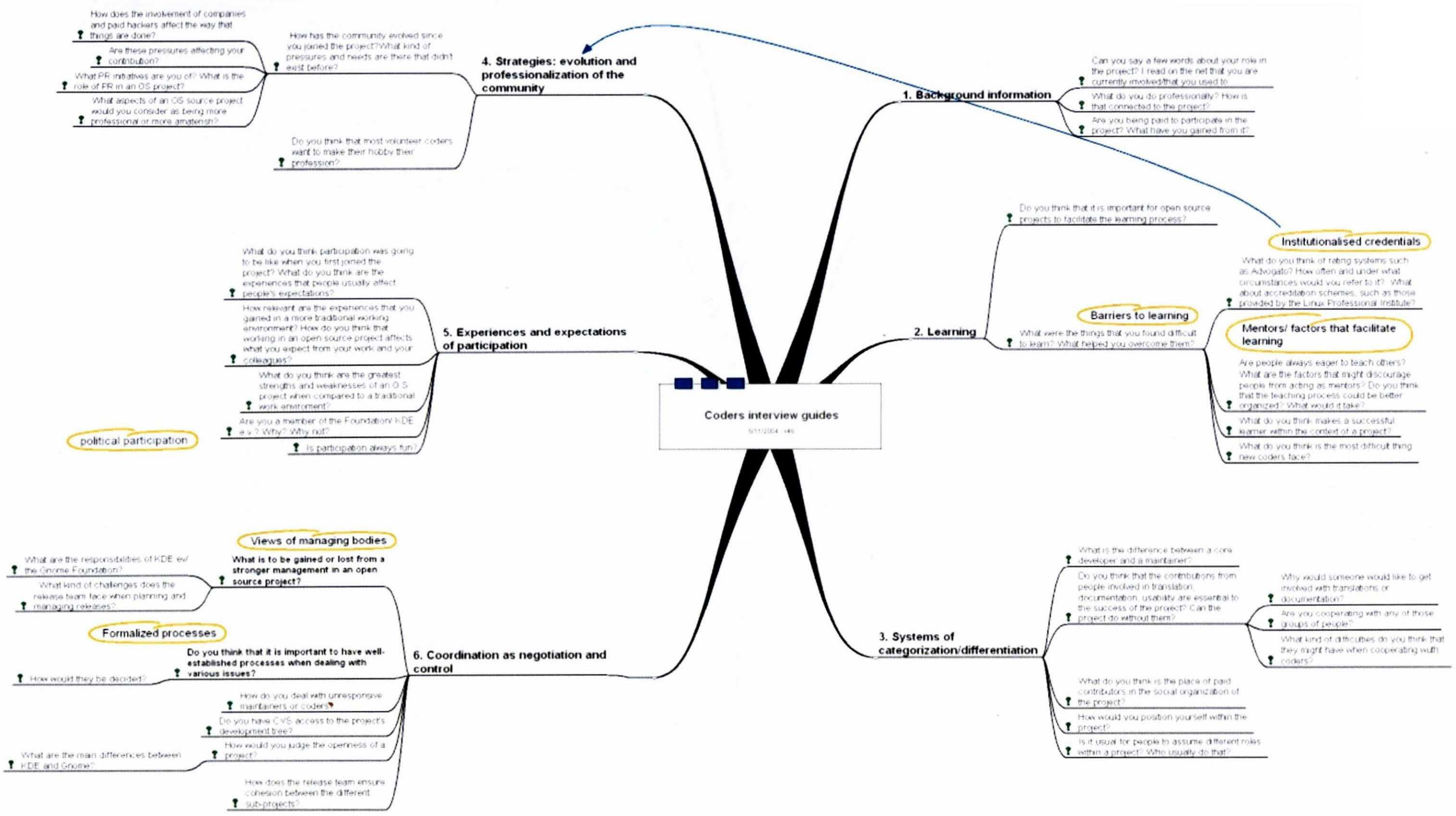
Zeitlyn, D. (2003). 'Gift Economies in the Development of Open Source Software: anthropological reflections.' Research Policy 32: pp.1287-1291.

Zhang, M. and J. Storck (2001). 'Peripheral Members in Online Communities.' Paper presented at the AMCIS 2001- the Americas Conference on Information Systems, Boston MA.

Zuboff, S. (1988). In the Age of the Smart Machine: The Future of Work and Power. New York: Basic Books.

Appendix

Qualitative Data



Coders interview guides

5/11/2004 - v46

1. Background information
   - Can you say a few words about your role in the project? I read on the net that you are currently involved/that you used to
   - What do you do professionally? How is that connected to the project?
   - Are you being paid to participate in the project? What have you gained from it?

2. Learning
   - Do you think that it is important for open source projects to facilitate the learning process?
   - Barriers to learning
   - What were the things that you found difficult to learn? What helped you overcome them?
     - Institutionalised credentials
     - What do you think of rating systems such as Advogato? How often and under what circumstances would you refer to it? What about accreditation schemes, such as those provided by the Linux Professional Institute?
     - Mentors/ factors that facilitate learning
     - Are people always eager to teach others? What are the factors that might discourage people from acting as mentors? Do you think that the teaching process could be better organized? What would it take?
     - What do you think makes a successful learner within the context of a project?
     - What do you think is the most difficult thing new coders face?

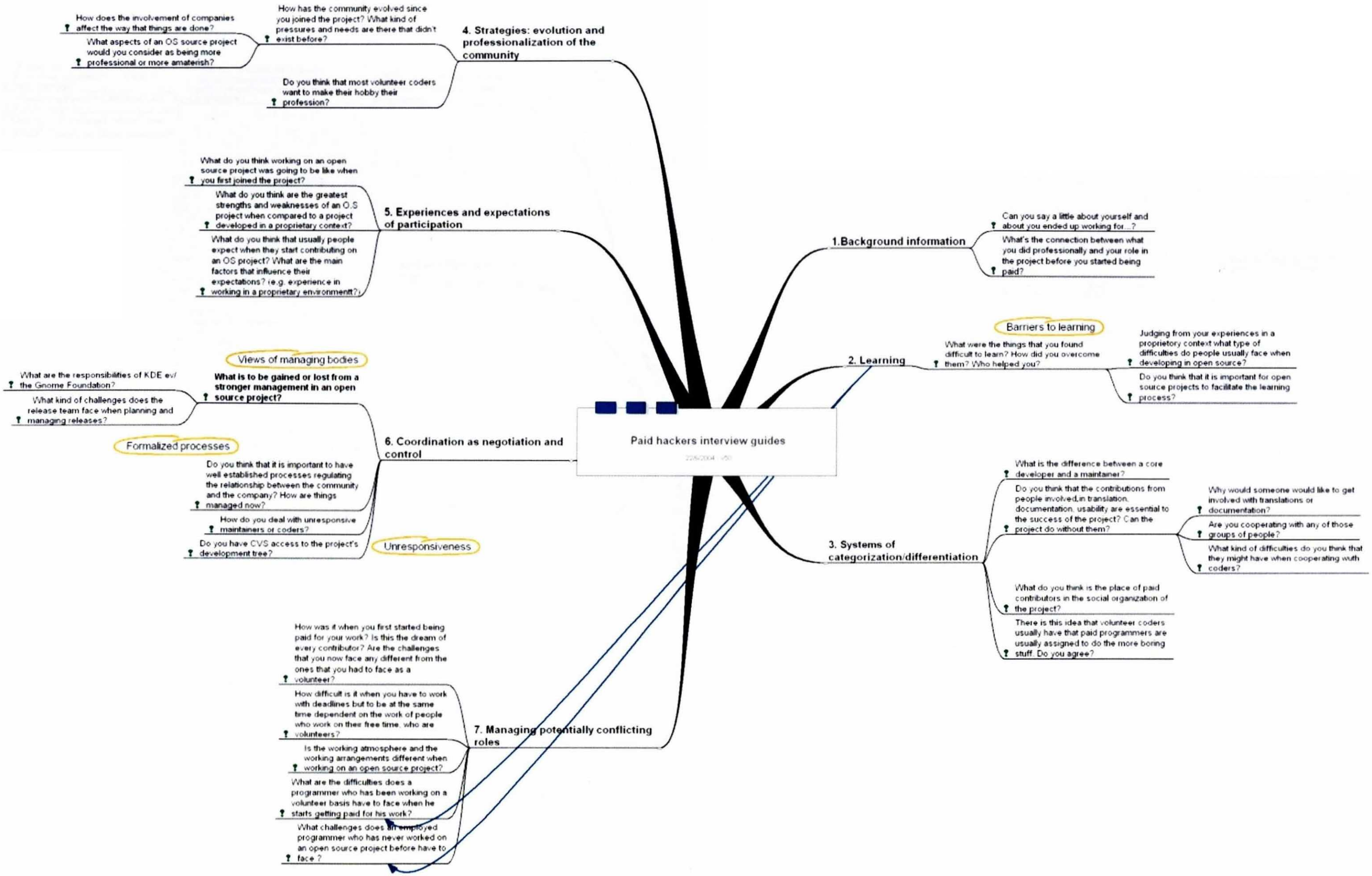
3. Systems of categorization/differentiation
   - What is the difference between a core developer and a maintainer?
   - Do you think that the contributions from people involved in translation, documentation, usability are essential to the success of the project? Can the project do without them?
     - Why would someone would like to get involved with translations or documentation?
     - Are you cooperating with any of those groups of people?
     - What kind of difficulties do you think that they might have when cooperating with coders?
   - What do you think is the place of paid contributors in the social organization of the project?
   - How would you position yourself within the project?
   - Is it usual for people to assume different roles within a project? Who usually do that?

4. Strategies: evolution and professionalization of the community
   - How has the community evolved since you joined the project?What kind of pressures and needs are there that didn't exist before?
     - How does the involvement of companies and paid hackers affect the way that things are done?
     - Are these pressures affecting your contribution?
     - What PR initiatives are you of? What is the role of PR in an OS project?
     - What aspects of an OS source project would you consider as being more professional or more amatenish?
   - Do you think that most volunteer coders want to make their hobby their profession?

5. Experiences and expectations of participation
   - What do you think participation was going to be like when you first joined the project? What do you think are the experiences that people usually affect people's expectations?
   - How relevant are the experiences that you gained in a more traditional working environment? How do you think that working in an open source project affects what you expect from your work and your colleagues?
   - What do you think are the greatest strengths and weaknesses of an O.S. project when compared to a traditional work environment?
   - Are you a member of the Foundation/ KDE e.v.? Why? Why not?
   - Is participation always fun?

political participation

6. Coordination as negotiation and control
   - Views of managing bodies
   - What is to be gained or lost from a stronger management in an open source project?
     - What are the responsibilities of KDE ev/ the Gnome Foundation?
     - What kind of challenges does the release team face when planning and managing releases?
   - Formalized processes
   - Do you think that it is important to have well-established processes when dealing with various issues?
     - How would they be decided?
   - How do you deal with unresponsive maintainers or coders?
   - Do you have CVS access to the project's development tree?
   - How would you judge the openness of a project?
     - What are the main differences between KDE and Gnome?
   - How does the release team ensure cohesion between the different sub-projects?



Paid hackers interview guides

22/6/2004 - v50

1.Background information

- Can you say a little about yourself and about you ended up working for…?
- What's the connection between what you did professionally and your role in the project before you started being paid?

2. Learning

Barriers to learning

- What were the things that you found difficult to learn? How did you overcome them? Who helped you?
  - Judging from your experiences in a proprietory context what type of difficulties do people usually face when developing in open source?
  - Do you think that it is important for open source projects to facilitate the learning process?

3. Systems of categorization/differentiation

- What is the difference between a core developer and a maintainer?
- Do you think that the contributions from people involved,in translation, documentation, usability are essential to the success of the project? Can the project do without them?
  - Why would someone would like to get involved with translations or documentation?
  - Are you cooperating with any of those groups of people?
  - What kind of difficulties do you think that they might have when cooperating wuth coders?
- What do you think is the place of paid contributors in the social organization of the project?
- There is this idea that volunteer coders usually have that paid programmers are usually assigned to do the more boring stuff. Do you agree?

4. Strategies: evolution and professionalization of the community

- How has the community evolved since you joined the project? What kind of pressures and needs are there that didn't exist before?
  - How does the involvement of companies affect the way that things are done?
  - What aspects of an OS source project would you consider as being more professional or more amaterish?
- Do you think that most volunteer coders want to make their hobby their profession?

5. Experiences and expectations of participation

- What do you think working on an open source project was going to be like when you first joined the project?
- What do you think are the greatest strengths and weaknesses of an O.S project when compared to a project developed in a proprietary context?
- What do you think that usually people expect when they start contributing on an OS project? What are the main factors that influence their expectations? (e.g. experience in working in a proprietary environmentt?)

6. Coordination as negotiation and control

Views of managing bodies

- **What is to be gained or lost from a stronger management in an open source project?**
  - What are the responsibilities of KDE ev/ the Gnome Foundation?
  - What kind of challenges does the release team face when planning and managing releases?

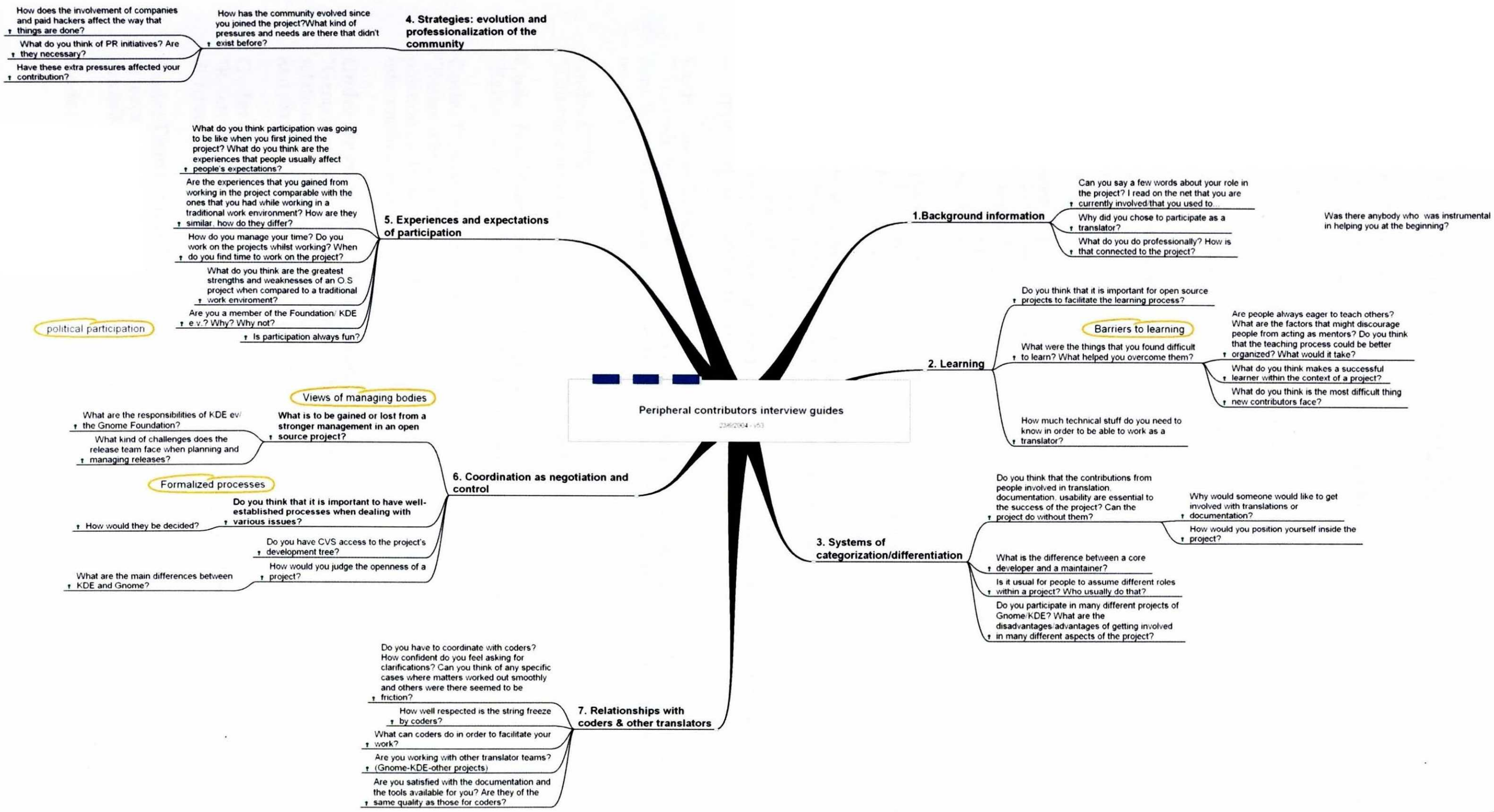
Formalized processes

- Do you think that it is important to have well established processes regulating the relationship between the community and the company? How are things managed now?
- How do you deal with unresponsive maintainers or coders?
- Do you have CVS access to the project's development tree?

Unresponsiveness

7. Managing potentially conflicting roles

- How was it when you first started being paid for your work? Is this the dream of every contributor? Are the challenges that you now face any different from the ones that you had to face as a volunteer?
- How difficult is it when you have to work with deadlines but to be at the same time dependent on the work of people who work on their free time, who are volunteers?
- Is the working atmosphere and the working arrangements different when working on an open source project?
- What are the difficulties does a programmer who has been working on a volunteer basis have to face when he starts getting paid for his work?
- What challenges does an employed programmer who has never worked on an open source project before have to face ?



Peripheral contributors interview guides

23/6/2004 - v53

1.Background information

- Can you say a few words about your role in the project? I read on the net that you are currently involved/that you used to...
- Why did you chose to participate as a translator?
- What do you do professionally? How is that connected to the project?

Was there anybody who was instrumental in helping you at the beginning?

2. Learning

- Do you think that it is important for open source projects to facilitate the learning process?
- Barriers to learning
- What were the things that you found difficult to learn? What helped you overcome them?
  - Are people always eager to teach others? What are the factors that might discourage people from acting as mentors? Do you think that the teaching process could be better organized? What would it take?
  - What do you think makes a successful learner within the context of a project?
  - What do you think is the most difficult thing new contributors face?
- How much technical stuff do you need to know in order to be able to work as a translator?

3. Systems of categorization/differentiation

- Do you think that the contributions from people involved in translation, documentation, usability are essential to the success of the project? Can the project do without them?
  - Why would someone would like to get involved with translations or documentation?
  - How would you position yourself inside the project?
- What is the difference between a core developer and a maintainer?
- Is it usual for people to assume different roles within a project? Who usually do that?
- Do you participate in many different projects of Gnome/KDE? What are the disadvantages/advantages of getting involved in many different aspects of the project?

4. Strategies: evolution and professionalization of the community

- How has the community evolved since you joined the project?What kind of pressures and needs are there that didn't exist before?
  - How does the involvement of companies and paid hackers affect the way that things are done?
  - What do you think of PR initiatives? Are they necessary?
  - Have these extra pressures affected your contribution?

5. Experiences and expectations of participation

- What do you think participation was going to be like when you first joined the project? What do you think are the experiences that people usually affect people's expectations?
- Are the experiences that you gained from working in the project comparable with the ones that you had while working in a traditional work environment? How are they similar, how do they differ?
- How do you manage your time? Do you work on the projects whilst working? When do you find time to work on the project?
- What do you think are the greatest strengths and weaknesses of an O.S project when compared to a traditional work enviroment?
- Are you a member of the Foundation/ KDE e.v.? Why? Why not?
- Is participation always fun?

political participation

6. Coordination as negotiation and control

- Views of managing bodies
- What is to be gained or lost from a stronger management in an open source project?
  - What are the responsibilities of KDE ev/ the Gnome Foundation?
  - What kind of challenges does the release team face when planning and managing releases?
- Formalized processes
- Do you think that it is important to have well-established processes when dealing with various issues?
  - How would they be decided?
- Do you have CVS access to the project's development tree?
- How would you judge the openness of a project?
  - What are the main differences between KDE and Gnome?

7. Relationships with coders & other translators

- Do you have to coordinate with coders? How confident do you feel asking for clarifications? Can you think of any specific cases where matters worked out smoothly and others were there seemed to be friction?
- How well respected is the string freeze by coders?
- What can coders do in order to facilitate your work?
- Are you working with other translator teams? (Gnome-KDE-other projects)
- Are you satisfied with the documentation and the tools available for you? Are they of the same quality as those for coders?

Code definitions-Thematic Analysis

Code: Areas of contribution
"This code refers to interviewees' or other individuals or groups of contributors areas of contribution in the project."

Code: Background
"References interviewees educational and professional background"

Code: Bugs
"This code includes quotes associated with bugs, such as reporting bugs, triaging bugs, solving bugs."

Code: Community and Project Evolution
"This code includes references to the way projects have evolved over time. Examples include additional demands placed by companies and the increase take up of the project by a wider user community."

Code: Community and Sociability
"This code includes references to social aspects of the community, such as contributors becoming friends and more generally developing an interest for the social aspects of the community (discussing issues related to common interests not related with developments, meeting offline for socializing, etc)."

Code: Companies and community
"This code includes specific and general references about companies and free/open source, their role and their involvement in projects."

Code: Control, Management and Agency
"This code includes references related to control, management & agency. More specifically it includes quotes related to issues of:
a. retaining/taking control and initiative
b. authority, applying pressure and influencing decisively the course of things.
c. stepping up and assuming a role or responsibility."

Code: Core Developers
"This code includes references to core developers. These are generally developers that contribute to fundamental aspects of development and have been involved in the project over a long period of time."

Code: CVS
"This code includes all references to CVS and CVS accounts."

Code: *Deciding* where to contribute
"References to how individuals decided to get involved in a specific area/module."

Code: Developers and Newbies
"References related to developers' helping out new contributors that reflect the their position and concerns in taking up this role. This code represents the view of the information providers."

Code: Developers and Peripheral Contributors
"General remarks regarding relationships between developers and non-coders, how their relations and workflows are structured, how they value each others' contributions, the gaps and overlaps that exist between their respective sphere of activities."

Code: Documentation
"References associated with finding, reading and producing project documentation at different levels of development."

Code: Finding a suitable task
"This code includes quotes about finding an appropriate task, usually for new contributors."

Code: Freezes

"References to freezes, different freeze phases, the challenges of maintaining freezes and the problems associated with breaking them. This code is associated with releases."

Code: Fun and Boredom
"This code includes references to fun and boredom."

Code: Getting into Gnome/KDE
"References to how interviewees became involved in GNOME and KDE"

Code: Getting into Linux
"References to how people came to know and use Linux"

Code: GNOME Foundation
"The code includes references related to the GNOME Foundation and the Foundation Board."

Code: Innovation and Maintainance
"The codes includes references associated with dynamics of innovation and maintainance, e.g. how people come up with new features, the diffuculties of maintaining a balance between the desire to be creative with the need to maintain a degree of stability."

Code: IRC channel
"References to IRC chat channels"

Code: KDE e.V.
"References to KDE e.V."

Code: Learning Process
"This code covers references related tolearning in free/open source and in particular:
a. what newbies and experienced contributors deem as being necessary, helpful or difficult during the process, such as for example specific resources and attitudes.
b. how they describe the process of gaining a better understanding of the project, solving problems and finding the information required to complete their objectives."

Code: Leaving and Staying
"Quotes related to showing consistency and commitment or of leaving the community, dropping responsibilities."

Code: Mailing Lists
The code includes references about mailing lists."

Code: Maintainers
The codes includes references related with maintainers."

Code: Managing boundaries firm(s) and community
The code includes references related to managing boundaries between companies and the community."

Code: Membership-Access
"This code includes references related to different aspects to membership and access such as the way they are defined and its different levels (for example being considered an 'honourary member') as well as various forms of inclusion and exclusion between and across different teams of contributors."

Code: Newbies
"This code includes quotes related to new contributors, their characteristics, common patterns of joining."

Code: Open Source and Professional Development
"This code includes references that involve:
a. How participation in volunteer projects has benefited participants in their professional life or is connected with their professional life (for example in terms of skills acquisition or a better understanding of the requirements of large scale software development).
b. Volunteers' desire to become employed to contribute to the projects of their choice.
c. How the manage the demands of their work life and their community contribution."

Code: Open Source and Proprietary Development
"This code includes refences that compare the Open Source Way with processes and dynamics of proprietary software development."

Code: Open Source and School
"Comparisons made between open source and learning/working at School/University, etc"

Code: Open Source and Work
"This code includes quotes comparing the open source way of cooperation and coordination to the dynamics of a traditional work environment."

Code: Organization, Processes & Rules
"This code includes quotations refering to sociotechnical and formal structural aspects and processes that relate with the organization of projects."

Code: Paid Developers
"This code includes quotes from and about paid developers and their role in the project. This code is associated with 'Companies and Community'.

Code: People-mentoring
"The code includes references related to the people as a problem solving resource and as mentors in the learning process. The quotes represent mainly the perspective of the help seekers."

Code: Peripheral contributors and their teams
"The code includes quotes related to:
a. the profile, motivation and characteristics of translators, documentation writers, usability experts and their team.
b. coordination among peripheral contributors.
c. the general importance of translations and other peripheral areas for the overall project."

Code: PR
"This code includes references related to Public Relations and Promotion related activities."

Code: Release
"The code includes references related to releases, more specifically how they are organized as well as how they structure other elements of participation."

Code: Resources
"The code covers:
community learning, information seeking and development resources that are not included explicitly in the other resource related categories."

Code: Self- Reliance
"The code includes references related to self-reliance, especially in relation to newbies and learning but also in connection to more general information seeking and problem solving."

Code: Task Lists
"This code refers to task lists, their existence or the need thereof."

Code: Technical Skills
"The code refers to:
a.the level of technical knowledge required from the part of peripheral contributors.
b. how different skill sets are appreciated within the community."

Code: The Open Source Way
"This code includes references to explicit and implicit values, characteristics and dynamics of the free/Open source process of development, such as how differences are resolved and decisions are taken, what it means to have the right attitude when asking and receiving feedback. and It is strongly associated with the codes: 'Open Source and Proprietary Development', 'Open Source and Work', 'Open Source and School', which are in effect its subsets."

Code: Time
"References to time and the lack thereof, i.e. finding time for development and assisting

other people."

Code: Unresponsiveness & Responsiveness
"Comments and remarks on how interviewees and other contributors can be responsive or unresponsive to different requests for help and they factors that may influence their stance, i.e. getting answers to queries quickly, etc."

Code: Users
"References to Users"

Phase 1: GNOME interviewees

Pseudonym	Profile	Date and type of interview	Roles/experience/ areas of involvement
1. Michael and Bill	*Michael is 31 and he is Belgian. He is one of the core developers of Gstreamer, a platform (library) for developing multimedia applications. At the time of the interview he had decided to leave his job and go work for Fluendo SL, a company based at Barcelona which will develop applications on top of Gstreamer. Bill is 22 and also develops on Gstreamer and GNOME. He had started to study Computer Science but dropped it because found it, compared with F/OS development, boring.*	Face-to-face interview at Fosdem, Brussels, 22/02/04	**Volunteer developers**/experienced
2. Sean	*Sean is 35 years old and lives in Wales. Although he is not a coder, he has been involved in the GNOME project from very early on. He has been triaging bugs, writing documentation and contributed to the internationalization effort.*	Phone interview, 09/03/04	**Peripheral contributor**/volunteer/experienced
3. Leroy	*Leroy is in his early thirties and works for Sun Microsystems Inc. developing GNOME. He got involved with GNOME before he sought to be assigned to work on it at Sun Microsystems Inc. His long standing experiences with the GNOME community has put him of the position of mediating between the teams at Sun and the team of developers at GNOME.*	Phone interview, 18/03/04	**Paid developer**/experienced
4. Dan	*Dan is 21 and studies Computer Science. He has been contributing to GNOME for a year now, mostly as a translator and documenter, although he occasionally submits small patches. He is the coordinator of the a translation team, and he is also involved in the Debian project.*	Phone interview, 06/04/04	**Peripheral contributor/** newbie coder/volunteer/ Translation team coordinator
5. Bruce	*Bruce is 27 and he is the head of the GNOME internationalization process. Although he knows how to program and has studied electrical engineering he has been mostly involved in translations. He also has administrative privileges. He can authorise new CVS and shell accounts.*	Face-to-face interview in Kristiansand, GUADEC, Norway, on 29/06/04 Follow up phone interview, 28/07/07	**Peripheral contributor/** coder/volunteer/
6. Al	*Al is in his early 40s. He has been recently been elected member of the Foundation Board. One of the reasons for pursuing this was to push the facilitate the access of third-party developers, coders who develop applications on the GNOME platform. Besides coding Al is also interested in documentation. In the company in which he works, most employees use GNOME as their default desktop.*	Phone interview, 24/06/04	**Volunteer developer**/experienced Member of the Foundation Board

7. Robin	*Robin is in his early thirties. He has been an important member of the release team for a number of years. He now considers his role as merely administrative. He argues that the release process has been so ingrained in community processes that it pretty much runs on autopilot. He is also involved in a number of promotional and advocacy activities.*	Face-to-face interview, GUADEC 2004, Kristiansand, Norway, on 28/06/04	**Volunteer developer**/experienced Release team coordinator
8. Ewan	*Ewan could be considered as an aspiring GNOME coder. He has been wanting to start contributing to the project for a while, but he his studies (he is in his early 20s) have not allowed him to emerge himself in the process. He believes that the community and the developers could do a better job keeping the sites and the documentation updated and integrated.*	Phone interview, 21/09/04	**Newbie**/volunteer coder
9. David	*David is 25 years old. He is French, leaves in the UK and works for RedHat Inc., although the two are not really related, since the work that he does for the company has nothing to do with his involvement in the project where he hacks on a movie player and works for the internationalization project.*	Face-to-face interview, Gilfold, London, 29/09/04	**Paid developer**/ experienced / also involved in translations
10. Martin	*Martin is one of the core contributors of GNOME. His contributors span many critical areas of the project. He is currently employed by RedHat Inc. where he works to improve various aspects of GNOME. He believes that the community has proven its professionalism and that certain formal processes are now redundant.*	Phone interview, 11/07/04	**Paid Developer**/experienced Core hacker
11. Neal	*Neal is 28, French and has a PhD. He has been involved in GNOME for almost two years now, by contributing patches here and there, but he still considers himself a newbie. He has been building his skills slowly and has only recently found the time to get more involved in the project. He has recently "inherited and started to co-maintain a small application.*	Phone interview, 19/10/04	**Newbie**/volunteer coder
12. Gerri	*Gerri is 24, French and has studied computer science. He has been involved in GNOME for a short time and his involvement mostly involves triaging bugs.*	Phone interview, 18/10/04	**Newbie**/volunteer coder
13. Flint	*Flint is Canadian. He has been lurking on the GNOME mailing list and the main development lists quite a while before he made his presence known. He has now started to contribute to a project that has been flagmarked as being appropriate for new developers, which according to him makes things considerably easier.*	Phone interview, 15/10/04	**Newbie**/volunteer coder

Phase 1: KDE interviewees

Pseudonym	Profile	Date and type of interview	Roles/experience/ areas of involvement
14. Larry	*Larry is in his early thirties. He is Dutch and a mature undergrad student.. He is a very committed promoter of KDE and although he started out by doing translations, he is now involved in more PR related activities, such as organizing KDE's participation in events and writing articles for KDE's main news site. He is now trying to set up a administrative body for the Dutch KDE community, that will allow him to scale up the group's promotional activities.*	Face-to-face interview, Fosdem , Brussels, 21/02/04	**Peripheral contributor**/volunteer/experienced

15. Dean	*Dean is a 30 year old British software developer based in Manchester. He has been contributing to KDE from its very beginning and has been involved in many different aspects of the project. For example, he has been developing some tutorials for new developers. He believes that the relationship between coders and translations is not problematic, mainly because their differences have now been worked out and there are processes and tools in place. He thinks that there are more contested relations, such as the one between the usability group and the rest of the developers.*	Phone interview, 22/03/04	**Volunteer developer**/experienced
16. Nick	*Nick is Dutch and 30 years old. He is the coordinator of the Dutch translating group. He is a cook by profession and he considers himself a "bad apple" amongst other contributors who whether they contribute by submitting code or not, appear to have a technical background. Although he now feels accepted and has started contributing in more technical areas (developing a database) when he first applied for a CVS account he did mention anything about his true profession.*	Phone interview, 02/04/04	**Peripheral contributor**/volunteer/experienced Translation team coordinator
17. Gordon	*Gordon is a 22 year old KDE hacker. He has been active in PR, mainly representing KDE in Linux related expos and has been involved in the British/English translation.*	Phone interview, 29/04/04	**Peripheral contributor/** newbie coder/volunteer
18. Sam	*Sam is 23 years old. He is American, based in Germany and works for one of the biggest software companies in Europe. He got his job there through his connections through the KDE community and part of his work agreement was to be able to spend some of his time hacking on KDE. He has been involved in KDE for four years and he considers himself to be a core developer. He primarily contributes code, but he has also been heavily involved in promotion. He is a member of KDE e.v.*	Phone interview, 14/04/04	**Volunteer Developer**/experienced Core hacker
19. Lawrence	*Lawrence is a Dutch software developer in his late thirties who has started to work on the Dutch translation team, because this provides a better use of his time.*	Phone interview, 05/08/04	**Peripheral contributor** & /experienced/coder
20. Alex	*Alex is a 27 year old Czech KDE hacker who is employed by SUSE Labs on a free sponsorship basis and who spends his time improving, optimizing or fixing various thing in KDE libraries.*	Face-to-face interview at KDE Akademy, Ludwigsburg, Germany, 22/08/04	**Paid developer**/experienced Core hacker
21. Sebastian	*Sebastian is a French software developer who has been involved in the KDE project almost from its very beginning. He has been employed to develop on KDE in the past and he does not mind doing the more boring stuff of development.*	Face-to-face interview at KDE Akademy, Ludwigsburg, Germany, 23/08/04	**Paid developer**/experienced Core hacker
22. Roger	*Roger is a core contributor of the KDE project. He is 28 years old and he lives in Germany. He is employed by Novell/SUSE to work on integrating KDE to its release. He assigns CVS accounts and moderates some of the central development lists. He is also the one who founded the KDE internationalization project.*	Face-to-face interview at KDE Akademy, Ludwigsburg, Germany, 22/08/04	**Paid developer**/experienced Core hacker

23. Ken	*Ken is from New Zealand and is 32 years old. He is heavily involved in documentation but spends quite some time with translations as well. He believes that translators and documenters are generally not interested in each other's work and that people who start contributing to documentation do not usually expect that the review process is as thorough as the one in coding.*	Face-to-face interview at KDE Akademy, Ludwigsburg, Germany, 24/08/04 Follow up phone interview, 01/10/04	**Peripheral contributor**/volunteer/experienced Documenter Core Contributor
Phase 2: Gstreamer/Fluendo interviewees			
Pseudonym	**Profile**	**Date and type of interview**	**Roles/experience/ areas of involvement**
1. Clint	*Clint is 28 year old Belgian developer who used to work for quite a long time as a volunteer in Gstreamer before joining Fluendo. His involvement in the project was spurred by his interest in media streaming. He has deep interest in processes of software collaboration and quality control and his role in Fluendo allows him to pursue this as an official part of his work. Clint believes that a more systematic development approach is essential in building a momentum, for getting more people involved, for proving that Gstreamer and Fluendo can deliver.*	Face-to-face interview in Barcelona and the Gstreamer/Fluendo summit, 19/02/05	**Paid Developer**/experienced
2. Jim	*Jim is a 30 year old Norwegian. He has been involved in F/OS for a number of years and before becoming employed by Fluendo was contributing on a volunteer basis to GNOME and Gstreamer. In Fluendo he is working on the company's business and marketing strategy. Management.*	Telephone interview, 14/02/05	**Peripheral Contributor**/Paid/ experienced
3. Paul	*Paul is a 31 year old developer finishing his PhD in computer science. He got involved in Gstreamer because he wanted to develop an application for making music. One of his primary concerns with Fluendo's involvement in Gstreamer is the company's bias towards video applications*	Face-to-face interview in Barcelona and the Gstreamer/Fluendo summit, 20/02/05	**Volunteer developer**/experienced
4. Michael	*Michael is a 31 years old Belgian developer working for Fluendo. He used to work on Gstreamer as a volunteer, but he abandoned the project for something more than a year, after a series of arguments with a few of the other developers. Now that he is back, Michael feels that he has to re-prove himself to the community.*	Face-to-face interview in Barcelona and the Gstreamer/Fluendo summit, 19/02/05	**Paid developer**/experienced
5. Victor	*Victor is a 27 year-old computer science student. He has been developing on Gstreamer as a volunteer for quite a long time. Victor is one of the hackers that have the most problem with the creation of Fluendo and he is the one of the two programmers behind the opposing design approach that the Fluendo hackers are trying to accommodate*	Face-to-face interview in Barcelona and the Gstreamer/Fluendo summit, 18/02/05	**Volunteer developer**/experienced
6. Jeremy	*Jeremy is a 31 years old programmer based in California. He has been developing in Gstreamer mostly as a volunteer. He got involved after he got laid off and found himself with a lot of time in his hands. He has created this consulting company that offers support for Gstreamer. He is mostly interested in using Gstreamer for streaming scientific* data	Face-to-face interview in Barcelona and the Gstreamer/Fluendo summit, 20/02/05	**Volunteer developer**/experienced

Phase 2: GNOME Bounty Hunt Contest interviewees

Pseudonym	Profile	Date and type of interview	Roles/experience/ areas of involvement
1. Tonni	*Tonni is a Canadian hacker employed by Ximian and now by Novell Inc. He used to be the Evolution maintainer and his task in the Bounty contest was to specify the Evolution related bounties. He argues that one of the reasons for including the Evolution bounties was to increase community development.*	Face-to-face interview in GUADEC 2005, Stuttgart, Germany, 30/05/05	**Paid developer**/experienced/ Bounty Hunt contest organizer
2. Harri	*Harri is in his late twenties and was of the core developers of GNOME and one of the founders of Ximian. He is also the person who came up with the idea of the bounty hunt and who set up the framework for the contest. He talks about the main challenges of the contest and on its potential to become an established development framework that complements volunteer labour.*	Face-to-face interview in GUADEC 2005, Stuttgart, Germany, 31/05/05	**Paid developer**/experienced -Bounty Hunt initiator
3. Burt	*Burt works for Novell and is based in Bangalore. He was mainly responsible for administering the second and third round of the contest. This involved: updating the websites in terms of what bounties have been claimed and which not, following discussions on mailing lists and on IRC and answering questions about the contest, liaising between the winners, the Foundation and the maintainers to transfer the money.*	Telephone interview, 03/06/05	**Paid developer**/experienced Bounty Hunt contest organizer
4. John	*John is a computer science student based in Holland. He has his own F/OS project and contributes marginally to other open source projects. He worked and claimed a bounty concerning newsgroups support in Evolution. He thinks that the contest is a great way for students to gain some money and points out that better work could have been done in publicizing it..*	Telephone interview, 20/06/05	**Paid developer**/experienced / Bounty Hunt contest winner
5. Alistair	*Alistair is a well-established member of the GNOME community who had some objections about the Bounty Contest., He rejects the idea of OS as a volunteer paradise and argues that there needs to be a balance between private and volunteer.*	Telephone interview, 13/06/05	**Ex paid developer**/experienced
6. Peter	*Peter is an American hacker based in California. He is quite experienced in F/OS development and he has dropped out of college to work for a well-known F/OS company e. He used to be the maintainer of the GAIM project and the bounty he claimed in the contest involved the integration of GAIM into Evolution.*	Telephone interview, 10/06/05	**Volunteer developer**/experienced/ Bounty Hunt contest winner

Phase 2: KDE Quality Teams interviewees

Pseudonym	Profile	Date and type of interview	Roles/experience/areas of involvement
1. Chris	*Chris is English and has just completed his philosophy degree at the university of Reading. He sees his involvement in KDE and F/OS software as an extension of his interest in political activism. The interview focuses on peripheral participation in general, on his involvement in promoting KDE within and outside the context of Quality teams. In the last part of the interview he talks about his plans to launch a new initiative to promote KDE that is modelled after the Spread FireFox project.*	Telephone interview, 22/06/05	**Peripheral contributor**/volunteer/experienced
2. Eric	*Despite being a programmer by profession Eric is involved in KDE's usability project. His interview focused on the difficulties of working with programmers and on the KDE Quality teams. Some of the difficulties involving providing advice to programmers on usability issues is that it's very difficult to communicate an idea about an application without actually doing it. The way most success stories in usability seem to go is that a usability expert takes an application that is completely done, create a new one and then spends the next couple of months trying to convince the developer to redo his application.*	Telephone interview, 10/08/05	**Peripheral contributor**/volunteer/experienced
3. Matt	*Matt is one of the lead contributors in the KDE documentation project. In addition to his involvement in the Quality teams, he also helps users through mailing lists and the IRC channel. Teaching people new things is one of the main reasons he got involved with writing documentation. He agrees that the free/open source world, including KDE, has fostered the idea that participating in a project is all about contributing code*	Telephone interview, 27/06/05	**Peripheral contributor**/volunteer/experienced
4. Steven	*Steven is the contributor behind the KDE Quality teams project. He has not a programming background but he has studied business administration and has been involved in free/software mainly out of ethical reasons. His desire to build a site where among other things newbies could find support derived mainly from his own experiences back when he started to work on documentation. In the interview he describes extensively his initial difficulties*	Telephone interview, 17/06/05	**Peripheral contributor**/volunteer/experienced

GUADEC

28. - 30. JUNE, 2004 KRISTIANSAND • NORWAY

Field Notes from GUADEC 2004-Kristiansand, Norway /27-29/06/2004

■ Sunday, 27/06/04

I arrived at Kristiansand and checked in at the hostel. As soon as I unpacked I got a text message from ?, one of the Gstreamer contributors who has joined the Fluendo company, asking me whether I would like to join the rest of the Fluendo team at the house that they had rented for some tacos. I said yes and he came to pick me up. Before we went to their place, however, we made a stop at the college campus where the conference was held. The team of Fluendo/Gstreamer was hard at work at the main lecture theatre, making sure that everything was ready for the following day. They were going to be responsible for streaming all the conference presentations and since this was this first official appearance of their company they wanted everything to be perfect. The group comprised of ? and his wife ?. I hadn't met ? before. He is the founder of the company. The others I knew since the previous GUADEC and we have had the chance of meeting at several other occasions (Fluendo, Barcelona).

After waiting for about two hours for the boys to finish, we (? and me) decided to go home and start preparing dinner. While ? was gone to buy some extra groceries, I discussed with ? the difficulties of being married to a hacker. "Hanging out with geeks, really makes you patient" she said. ? returned with the extra groceries, but took off almost immediately because he had to go and pick up some people from the airport. He, nevertheless, found time to give us some instructions for dinner. We started discussing with ? the situation of "women stay home cooking, while men do all the coding".

The developers arrived at around 11. ? arrived shortly after with ?, ? was going to stay at their place. Funny comments about the fact that half of developers that arrived at Kristiansand hadn't made any arrangements for accommodation. No alcohol whatsoever was served during dinner. I don't know whether they normally do that, or whether they didn't want to drink anything because they had to get up early the following moming to finish the set-up. ? comment's about being high on adrenaline after an intensive day's work. Everybody was teasing ? who works for Red Hat in London. Apparently he also does some work for KDE. ? asked me who I was. I said that I was a student examining the social organization of open source and tried to avoid giving any further details, but people are curious about me (being one of the few women around) and I think that it will be difficult to keep a low profile. This is wasn't as much a problem in Fosdem. Drive back to the hostel.

■ Monday, 28/07/04-First day of the conference

Morning: I arrived at Adger campus and skipped the long queue at registration, since I had already registered and picked up my conference pack the day before. The welcoming speech was given by Tim Ney (Gnome's Foundation Executive Director) and Nat Friedman (Ximian's founder and currently Novell's vice president) and was more low key than last year's. After Tim thanked the sponsors and the organizers of the conference (he referred to the Fluendo people as the new kids on the block and thanked them for streaming the conference), Nat asked everybody in the room to stand up and say a few words about who they are and what they were doing in the project. That caught me by surprise! I was seating in the third row and I thought that if that moment I stood up and left it would be much worse that staying and say something about myself. Lying or saying something really vague, like, I am student, or a Gnome user, wouldn't work either, since there were several people in the room, who already knew that I was studying the project. So, when my tum came, I said that I was a student who was doing a study of the social organization of open source projects.

That got me thinking about my initial idea of treating these conferences as an opportunity for doing participant observation. That has been increasingly problematic, since most women that are present in these events are either developers/contributors or developers' girlfriends and I can't pose as either of them. Moreover, most of the times I feel that I am as much being observed than observing. When you are one of the few women attending these events people are bound to be curious and when you appear secretive, they will always assume the worst. In addition, given the fact that several people knew why I was there, and the likelihood of discussing this with others, I didn't want to appear inconsistent (me saying one thing, and them saying another).

After "coming out" to the GNOME community I managed to locate ?, the translators' coordinator who I wanted to interview. We agreed on doing the interview the following day. I met Femando San Martin Woerner. Femando was going to do a presentation on the conference's last day (Wednesday, the 30th of June) about SMEs and open source software in Chili drawing primarily from his experiences as an IT manager in a Chilean estates company that has started developing their own open source IPR applications. Femando's name had come up in an email exchange with Tim Ney about the DBE project. Since I would not be able to see his presentation (I had to fly to Brussels for the DBE meeting on Wednesday) Femando was kind enough to outline the main points of his presentation for me. He mentioned that what he was going to emphasize was "the experience of the inexperience" of Chilean SMEs in regards open source development. The abundance of pirated Microsoft software and the gap between the commercial programmers community and the hacker community means that it is very difficult for open source development to scale up in an SME context. He said that he hoped that something could be done if SMEs could group up and commission open source applications collectively.

He promised to forward me the slides from his presentation and I said that I would try to forward to him whatever material I found interesting from the DBE.

12:30: I started the interview with ?, the release team manager but we had to stop short since both of us wanted to attend ? presentation on innovation within open source. ? seems to possess all the characteristics of a hacker enfant extraordinaire. He graduated from MIT having studied computer science and mathematics. Back in 1999 he co-founded with Miguel de Icaza Ximian, a company that was created as a spin-off of the GNOME project. He is now the vice-president of Research and Development of Novell, that bought Ximian in 2003. After going over GNOME's history by using screenshots from different periods of the project's development, Nat concentrated on innovation in regards visualization techniques, to illustrate how many things can still be done. He also made a few important points about innovation within the open source movement and the factors that stifle it. He suggested that the free/open source culture has become intolerable to newbies and that the barriers to entry have become too high. With the exception of lead developers, people and especially new developers, out of fear of embarrassment, are too intimidated to innovate. He then presented Beagle, a very interesting search tool, that is built upon some of the visualization techniques that he had previously mentioned.

The other interesting presentation of the day was given by Tim Foster, head of the Sun Software Globalization team. Tim explained the process behind translation within Sun. It appears that there is some degree of cooperation between companies and communities even at a translation level. This is an issue that I had completely overlooked. Tim presented some of the tools that have been developed for the translators within Sun and there was an interesting discussion about how the community could take advantage of them. It appeared that the only thing that prevented this was a technical issue, but I doupt that this is the case. It appears that there isn't a unified database for all translations within GNOME, since the project is organized in language teams, each with its own CVS project and the Sun application works by cross-checking translations from a unified database (for more details, see Tim's article in the conference proceedings). Sun supports only 9 languages but they use community translations for the other languages (GNOME is translated in around 50) although they stress that they don't support them. Although they have currently one person who is responsible for the supported languages, they outsource the actual translating work. The Russian translators coordinator complained about the effectiveness of communication between the community and the translators from third party companies. From what I understood, before Sun had appointed people from within the company to act as mediators between the third party translator companies and the community there have been considerable misunderstandings, for example community translators would not be consulted by third party translators upon the use of terminologies, with the result that a lot of work was duplicated and that the quality of the translation became questionable.

After the presentation I had a chat with ?. She told me that she was a bit weary of the adoption of Sun's translation tool, since it meant that translators would lose their access to CVS. ? always has some fantastic insights to offer on the workings of F/OS.

Before that I went for a walk with ? and ? from Sun. That was the only sightseeing that I had time to do. ? was really excited about an email that he had received from a guy at Sony who is experimenting on using Gstreamer for some of their products. A few days ago there was a rumour, which was confirmed by Tim in his introductory speech, that the Real Player was going to be released under an OS license and that has been a real blow for the Fluendo people (since it meant that there was now a very reliable rival open source media player). ? also has some contacts within the BBC that appear to be interested in Gstreamer.

Back to the conference and ? was nowhere to be seen. I had to get hold of him to finish the interview. Maybe I will have to settle with finishing it over the phone. Saw ? and ? and decided to go home. While waiting for the bus, I started talking to this girl who I had previously noticed at the conference. She looked just as a girl hacker should look [desciption]. Her name was ?. Before long she invited me to have dinner with her friends at a Mexican restaurant down town and I accepted. In the bus I realized that I was going to have dinner with Gimp's core developers (Gimp is one of the most popular open source manipulation programs that many consider as powerful as Adobe's Photoshop http://www.gimp.org/). On the bus, I was introduced to ?, ? and ?. When we arrived at the restaurant ? mentioned proudly that we were going to be joined by ? and ?, otherwise known as barefoot ?. Had to suppress a chuckle when I said that I already knew ?, ? seemed to be surprised. ? is also a well-known figure within the movement. Besides being a militant vegetarian, ? never wears shoes but seems to overcompensate by wearing ridiculous hats. He works for the World Wide Web Consortium.

? was interested in how I got into open source. I explained that it started with a small essay for a Master's degree course, that shortly afterwards I decided to do my dissertation on Linux users and that before long I was doing my PhD research on it. She was curious about my findings, but I told her that I haven't done a proper analysis yet and that she would have to wait until October.

Had an interesting discussion with ? about her work with Gimp. She said that she was convinced that women can't code, but that they can inspire men to program. She was really frustrated with someone from within GNOME who had taken down the Gimp website on which she was working for quite a while and who managed to break it. She couldn't believe that 6 months worth of work was destroyed by someone who had a CVS account, but "who couldn't run Apache". Some interesting comments about the technical expertise of people with CVS accounts, about how easy it was to acquire one at the early stages of the project and how difficult it is get one nowadays.

■ Tuesday, 29/06/04-Second day of the conference

Morning: Coffee with ? and the Gimp crowd.

I attended the talk by Liam Quin on design for programmers. He outlined some basic design principles for programmers, especially those that design user interfaces. I always felt that certain dialogue boxes were disastrously designed and now I know why! Lunch with ?, no Fluendo people around. Apparently they took the day off, since everything appeared to be running smoothly. ? mentioned that he had moved to Barcelona, after resigning from his job at the bank in Brussels. ? also resigned from his position as a Senior Consultant in the Norwegian division of Oracle. Everybody seems to be investing a lot in the company. I ran into ? and we arranged to do a phone interview sometime next week. I should play back the existing interview to see what we have already covered.

I talked to ? about the progress of my work and asked him if he knew anything about the Bounty Hunt, one of the nested case studies that I intend to do. I had considerable trouble finding out who was involved in its organization (although the names of the winners have been published, the names of organizers were not mentioned). He asked me why I was interested in it. I said that it appears to cover a need that is not being addressed either by the volunteers or the paid developers. I think that he found my justification satisfactory. He mentioned that the contest has been somewhat controversial, since that are fears that it might create a gold rush within the community. He added that, although, the cash bounties are not very significant for American developers, they are for hackers from Eastem European Countries. He also mentioned something about IBM having a patent about paying in open source. I am not sure what he meant, but I need to follow it up.

The person that ? pointed me to about the Bounty Hunt is ?. Although Nat Friedman originally conceived the contest ? has been responsible for setting up the first round. ? mentioned a few other names involve in it at an organizational level, namely ? who was responsible for specifying the contest's rules, ? responsible for the mockups and ? currently in charge of the contest's second round. He works at the Bangalore division of Novell. I met ? briefly and he gave me his mobile number. ? also offered to help me find any other relevant material, since some stuff related with the competition, especially about its first phase have been taken off-line.

I also talked to ?, project manager in Wipro. She asked me to call/email her to suggest a few people from her team that have been involved in GNOME. They have been working with Sun on open source projects for about 2,5 years.

I had an interesting discussion with ? from Sun, whom I have already interviewed. I recorded part of our discussion, since the current project that ? is working on is really interesting. ? is trying to get Sun and the community to synchronize their review processes. The main problem at the moment is that the results of Sun's review of GNOME's interfaces cannot be shared with the community, since internal regulations classify them as confidential. One the other hand, the community's documentation is not being regularly updated and there are many issues about the way (if any) that the community implements the suggestions of companies. It seems that Sun is willing to share its reviews, if the community will provide some assurance that some of their demands will be addressed on a timely basis (before each major release). ?

was satisfied because he had got some good suggestions as to how he should proceed to introduce the issue to the community. He initially thought to directly address the developers, but he was advised against it. The rationale being that developers are not generally interested in documentation and that they would probably consider the issue insignificant. Instead he was advised to put the request through GNOME's advisory board, which consists of representatives of the main companies involved in GNOME.

? intended to put his request to the Advisory board which would then appeal to the Foundation. After reaching a, hopefully, positive decision, the Foundation would then present their decision to the developers. I am not entirely sure how binding this decision would be or how it would be presented, but ? was really excited since he believed that he was the first person to take advantage of a process that had apparently never been used.

At the barbeque which was organized by the conference organizers, ? introduced me to ? a Sun developer that has been involved in GNOME for quite a while. Arranged an interview. Went to the bar. Exchanged email addresses with the Gimp crowd. Met ?. There was a Foundation meeting earlier and he appeared to be satisfied with how it went. ? bug master at Ximian joined us and asked me how my dissertation was going. He said to ? that he was impressed with the consideration that the Fluendo people had given to licensing issues. He advised him to think also through to the boundaries ("lines") that they would need to put in place between the community and the company ("corporate"). He mentioned the case of the Evolution developers, "who were thinking community, community, community and then one day they woke up and they realized, that, shit! I am corporate". The earlier these lines are drawn according to ?, the less trouble the Fluendo people will have in negotiating their double identity. "What are you going to do" asked ? "if your clients push you to deliver a product?".

? also commented on how difficult it has been for him after Ximian's takeover by Novell. He was no longer involved in the important decision making. He mentioned some of the problems Novell had when they acquired Suse (Suse uses KDE and not GNOME as their default desktop). He said that the differences Novell had with Suse mirrored the differences between GNOME and KDE. We are more pragmatic, he said. We didn't want to be small anymore and it has been more important for us to make GNOME the default Linux desktop. After the takeover Suse fought all the way to keep their lizard logo, and we changed ours within a weekend.

Said goodbye to ? and promised ? to go and visit them at Barcelona as soon as I had the chance. I should combine it with the interviews for the Fluendo/Gstreamer study. Most of the Gstreamer people would be at the KDE conference in Stuttgart at the end of August, so we would be able to organize the details.

Quantitative Data

Additional data are available upon request

Dear «First Name»,

* Answering these 6 questions requires only one minute of your time and is really important for my research. They form part of a Social Network Analysis study that I am conducting as part of my PhD at the London School of Economics and they complement interviews that I have been conducting with Gnome contributors for more than a year now. The results will be made available on the web.

*The collected data will remain anonymous.

*Participating in the survey means that you are eligible for a draw for a 75$ gift voucher from amazon.com! The name of the lucky respondent will be announced on the Gnome Foundation List.

*For more information regarding the specific aims of the study and the more general framework of my research go to: http://mail.gnome.org/archives/foundation-list/2005-June/msg00015.html

---

Questions:

1. Area(s) of contribution:

From the Gnome Foundation membership list I was able to see that you are involved in the following area(s) of development: «Areas of Contribution».

a. Is this accurate? Yes/No

b. If this information is not accurate, please list the area(s) of your contribution:

Area(s):

c.If you are contributing to more than one modules or areas could you please indicate the module/area that you spent the most time working on during the last SIX months

Main:

2. Please indicate whether you are employed by anyone to work on some aspect of Gnome development (toolkit/platform/applications/documentation, etc.):

----

a. Employed/contracted: Yes/No

If yes, please specify the name of your employer:

b. Name of employer:

----

a.Guadec 2003-Dublin: Yes/No

b.Guadec 2004-Kristiansand: Yes/No

c.Guadec 2005-Stuttgart: Yes/No

5. Where do you currently live? (city, country)

Answer:

6. Is there something else related to these questions that you would like to add?

Answer:

That was it! Thanks :)

Evangelia Berdou
e.berdou@lse.ac.uk
«ID number»

Dear «First Name»,

* Answering these 6 questions requires only one minute of your time and is really important for my research. They form part of a Social Network Analysis study that I am conducting as part of my PhD at the London School of Economics and they complement interviews that I have been conducting with KDE contributors for more than a year now. The results will be made available on the web.

*The collected data will remain anonymous.

*Participating in the survey means that you are eligible for a draw for a 75$ gift voucher from amazon.com! The name of the lucky respondent will be announced on the KDE e.V. list.*For more information regarding the specific aims of the study and the more general framework of my research see the attached document.

Questions:

1. Area(s) of contribution:

----

a. Please list the areas of your contribution to the KDE:

Answer:

b.If you are contributing to more than one module or area could you please indicate the module/area that you spent the most time working on during the last SIX months

Main:

2. Please indicate whether you are employed by anyone to work on some aspect of KDE development (toolkit/platform/applications/documentation, etc.):

----

a. Employed/contracted: Yes/No

If yes, please specify the name of your employer:

b. Name of employer:

3. Please indicate whether you are currently employed by anyone to work on OTHER parts of free/open source software:

----

a. Employed: Yes/No

If yes, please specify the name of your employer:

b. Name of employer:

4. Please indicate whether you have you attended the following KDE Community World Summits:

----

a. KDE aKademy 2003-Kastle Nove-Hardy:

b. KDE aKademy 2004, Ludwigsburg:

5. Where do you currently live? (city, country)

Answer:

6. Is there something else related to these questions that you would like to add?

Answer: That was it! Thanks:)Evangelia Berdoue.berdou@lse.ac.uk

Modules included in the quantitative examination of patterns of maintainership

GNOME 2.10 release

	Platform Sources		**Desktop Sources**		**Desktop Sources**
no	*Module*	*no*	*Module*	*no*	*Module*
1	GConf	39	Bug buddy	82	gnopernicus
2	ORBit2	40	**Control center**	83	gok
3	at-spi	41	Dasher	84	gpdf
4	atk	42	Eel	85	gst plugins
5	atk	43	Eog	86	**gstreamer**
6	*audiofile*	44	Epiphany	87	gtk-engines
7	esound	45	**Overall Evolution Maintainer**	88	gtkhtml
8	gail	46	**Evolution Mail**	89	gtksource view
9	glib	47	**Evolution Addressbook**	90	gucharmap
10	gnome mime data	48	**Evolution Calendar**	91	*libgail-gnome*
11	gnome vfs	49	**Evolution Composer**	92	libgtkhtml
12	gtktreeview	50	**Evolution EPlugin**	93	libgtop
13	gdk-pixbuf	51	**EvolutionGroupWise**	94	librsvg
14	gdkuiManager	52	file roller	95	libsoup
15	*Gobject*	53	gal	96	libwnck
16	*Win32 port of GTK*	54	gcalctool	97	libxklavier
17	GTKFileChooser	55	**gconf-editor**	98	**metacity**
18	*GTKComboBox*	56	**gdm**	99	**nautilus**
19	*GtkToolbar*	57	gedit	100	nautilus CD burner
20	Pango	58	ggv	101	nautilus media
21	X11 port of GTK	59	gnome applets	102	scrollkeeper
22	Core Widget System	60	gnome backgrounds	103	soundjuicer
23	*Glib thread support*	61	**gnome-desktop**	104	start-up notification
24	Intltool	62	gnome doc-utils	105	system tools backends
25	libIDL	63	gnome games	106	totem
26	*libart*	64	gnome icon theme	107	vino
27	libbonobo	65	**gnome keyring manager**	108	vte
28	*libbonoboui*	66	gnome mag	109	ximian connector
29	libglade	67	**gnome media**	110	yelp
30	libgnome	68	**gnome menus**		
31	libgnome canvas	69	gnome netstatus		
32	libgnome print	70	gnome net tool		
33	libgnome printui	71	**gnome panel**		
34	libgnomeui	72	**gnome session**		
35	libxml2	73	gnome speech		
36	libxslt	74	gnome system monitor		
37	pango	75	gnome system tools		
38	pkgconf	76	**gnome terminal**		
		77	gnome themes		
		78	gnome utils		
		79	**gnome volume manager**		
		80	gnome2 user docs		
		81	gnome meeting		

KDE 3.4 Release

	KDE LIBRARIES		**KDE BASE**		**KDE multimedia**
no	subproject	no	subproject	no	subproject
1	Arts	39	drkonqi	75	KDE multimedia framework
2	Dcop	40	kappfinder	76	arts
3	kab	41	kate	77	kfile-plugins
4	kabc	42	kcontrol (framework)	78	mpeglib
5	Kate-libs	43	kdcop	79	noatun
6	kcert	44	kdebugdialog	80	aktion
7	Kcmshell	45	kpasswd	81	kaudiocreator
8	kconf_update	46	kdeprint	82	kaboodle
9	kdecore	47	kdesktop	83	kmid
10	kdoctools	48	kdesu	84	kmidi
11	khtml	49	kdialog	85	kmix
12	kmdi	50	kdm	86	kscd
13	knewstuff	51	khelpcenter	87	krec
14	kparts	52	khotkeys		

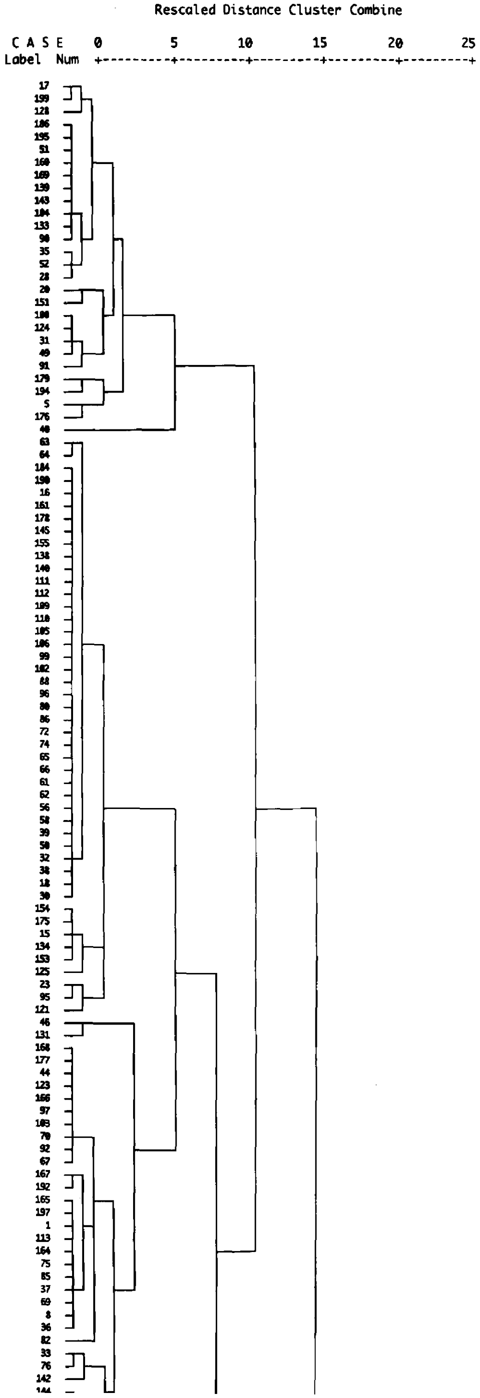
15	kresources	53	kicker		
16	kspell 2	54	kioslave		
17	kstyles	55	klipper		
18	kunnitest	56	kmenuedit		
19	kdeprint	57	konqueror		
20	kded	58	konsole		
21	kdefx	59	kpager		
22	kdesu	60	kpersonalizer		
23	kdeui	61	kreadconfig		
24	kdewidgets	62	krichtexteditor		
25	kimgio	63	kscreensaver		
26	kinit	64	ksmserver		
27	kio	65	ksplashml		
28	kio/kfile	66	kstart		
29	kio/kssl	67	ksysguard		
30	kioslave http	68	ksystraycmd		
31	kioslave file	69	ktip		
32	kioslave ftp	70	kwin		
33	kjs	71	kxkb		
34	kwallet kwalletbackend and k	72	l10n		
35	libkmid	73	libkonq		
36	libkscreensaver	74	nsplugins		
37	libltdl				
38	mimetypes				
	KDE PIM		**KDE network**		**KDE webdevelop**
no	*subproject*	*no*	*subproject*	*no*	*subproject*
88	certmanager	122	kdict	162	Quanta
89	kaddressbook	123	kit	163	Kommander
90	kalarm	124	kdnssd	164	kmousetool
91	kalarmd	125	knewsticker	165	kmagnifier
92	kandy	126	kpf	166	kmouth
93	karm	127	kppp	167	kdevelop
94	kdgantt	128	ksirc	168	KFileReplace
95	kgantt	129	ktalkd	169	KXSLDbg
96	kioslaves	130	kxmlrpc	170	KImageMapEdit
97	kitchensync	131	lanbrowsing	171	KLinkStatus
98	kmail	132	krfb		**KDEdsdk**
99	kmailcvt	133	krdc	172	kapptemplate
100	kmobile	134	kwifimanager	173	kbabel
101	knode	135	kopete	174	kbugbuster
102	knotes		**KDE utils**	175	kdepalettes
103	konsolekalendar	136	ark	176	kmtrace
104	kontact	137	karm	177	kspy
105	korganizer	138	kcalc	178	kstartperf
106	kpilot	139	kcharselect	179	scripts
107	ksync	140	kdepasswd	180	scheck
108	ktnef	141	kdessh	181	kuiviewer
109	libical	142	kdf	182	umbrello
110	libkcal	143	kedit	183	kpovmodeler
111	libkdenetwork	144	kfind	184	kruler
112	libkdenetwork/scoring	145	kfloppy	185	ksnapshot
113	libkdenetwork/kpgp	146	kfontmanager	186	Kpdf
114	kfoldertree, ksubscription,	147	kgpg	187	kuickshow
115	cryptplugwrapper	148	khexedit	188	kview
116	libkdepim	149	kjots	189	kviewshell
117	addresseeview, kvcarddrag	150	klaptopdaemon	190	libkscan
118	addressesdialog, komposer	151	klipper		**KDEaccessibility**
119	maillistdrag	152	kljettool	191	KDEaccessibility
120	libkpimexchange	153	klpq		
121	libksieve	154	ksim		
		155	ktimer		
		156	kwalletmanager		
		157	knotes		
		158	kpm		
		159	ktimemon		
		160	ktop		
		161	kwrite		

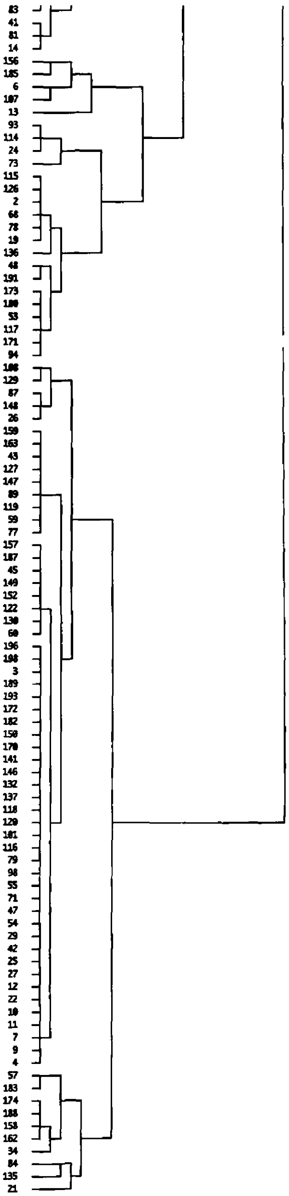
Organizations sponsoring GNOME Foundation Respondents

	Organization	Paid to contribute to GNOME	Paid to contribute to GNOME and F/OS	Paid to contribute to F/OS	Total
1	AB Take it		1		1
2	Async Open Source Ltda.		1		1
3	Baum Retec AG	1			1
4	Canonical LTD		3	1	4
5	Fluendo SL	2	1		3
6	Golder Software Systems Ltd.			1	1
7	Hewlett Packard Inc.		1		1
8	Hispafuentes SL			1	1
9	IBM Corporation			1	1
10	Ildana Teo			1	1
11	Imendio AB	1	2		3
12	Lamdaux-AUX Software services			1	1
13	Mandriva, Inc.		1		1
14	Molinux		1		1
15	Movial Corporation	1	1		2
16	Nokia Corporation		1	1	2
17	Novell, Inc.	9	6	1	16
18	OpenedHand Ltd.	1			1
19	OpenSource DT[redacted]	0	1		1
20	Red Hat, Inc	8	5	3	16
21	Sun Microsystems, Inc [redacted]	7	5		12
22	rpath, Inc.			1	1
23	Scalix, Inc.	1			1
24	Sharif Farsiweb, Inc.	1	2		3
25	sysref		1		1
26	Telsidel		1		1
27	Høgskolen i Agder and Aust-Agder Kulturhistoriske Senter			1	1
28	Unitet A.S.			1	1
29	Multitel			1	1
30	IFIS-Passau			1	1
31	Concordia University of Alberta	1			1
32	St-Antonius College Gouda			1	1
33	University of Grenoble 1			1	1
34	University Siegan			1	1
35	Vrije Universiteit Amsterdam			1	1
36	Vmware, Inc.	1			1
37	WEC			1	1
38	Wild Open Source, Inc.			1	1
39	Wipro Technologies, Ltd.	2			2
40	X-tend NV			1	1
41	Self-Employed/Subcontracting work	2	1	5	8
	Total	38	35	28	101

GNOME Foundation RESPONDENTS DENDROGRAM

Dendrogram using Complete Linkage




83
41
81
14
156
185
6
107
13
93
114
24
73
115
126
2
68
78
19
136
48
191
173
180
53
117
171
94
108
129
87
148
26
159
163
43
127
147
89
119
59
77
157
187
45
149
152
122
130
60
196
198
3
189
193
172
182
150
170
141
146
132
137
118
120
101
116
79
98
55
71
47
54
29
42
25
27
12
22
10
11
7
9
4
57
183
174
188
158
162
34
84
135
21

Frequency Tables of GNOME respondents areas of contribution -Low Level Coding

Component model

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	not contributing	189	95.0	95.0	95.0
	contributing	10	5.0	5.0	100.0
	Total	199	100.0	100.0	

Accessibility

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	not contributing	195	98.0	98.0	98.0
	contributing	4	2.0	2.0	100.0
	Total	199	100.0	100.0	

Graphical libraries

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	not contributing	183	92.0	92.0	92.0
	contributing	16	8.0	8.0	100.0
	Total	199	100.0	100.0	

other core modules

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	not contributing	183	92.0	92.0	92.0
	contributing	16	8.0	8.0	100.0
	Total	199	100.0	100.0	

Core desktop components

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	not contributing	173	86.9	86.9	86.9
	contributing	26	13.1	13.1	100.0
	Total	199	100.0	100.0	

Evolution

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	not contributing	183	92.0	92.0	92.0
	contributing	16	8.0	8.0	100.0
	Total	199	100.0	100.0	

Multimedia framework

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	not contributing	188	94.5	94.5	94.5
	contributing	11	5.5	5.5	100.0
	Total	199	100.0	100.0	

Secondary libs

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	not contributing	194	97.5	97.5	97.5
	contributing	5	2.5	2.5	100.0
	Total	199	100.0	100.0	

Secondary desktop components

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	not contributing	180	90.5	90.5	90.5
	contributing	19	9.5	9.5	100.0
	Total	199	100.0	100.0	

User applications

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	not contributing	143	71.9	71.9	71.9
	contributing	56	28.1	28.1	100.0
	Total	199	100.0	100.0	

Integration and porting work

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	not contributing	191	96.0	96.0	96.0
	contributing	8	4.0	4.0	100.0
	Total	199	100.0	100.0	

general bug fixes

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	not contributing	183	92.0	92.0	92.0
	contributing	16	8.0	8.0	100.0
	Total	199	100.0	100.0	

Development tools (bugzilla)

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	not contributing	167	83.9	83.9	83.9
	contributing	32	16.1	16.1	100.0
	Total	199	100.0	100.0	

language bindings

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	not contributing	192	96.5	96.5	96.5
	contributing	7	3.5	3.5	100.0
	Total	199	100.0	100.0	

release planning

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	not contributing	194	97.5	97.5	97.5
	contributing	5	2.5	2.5	100.0
	Total	199	100.0	100.0	

Packaging

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	not contributing	186	93.5	93.5	93.5
	contributing	13	6.5	6.5	100.0
	Total	199	100.0	100.0	

Usability

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	not contributing	194	97.5	97.5	97.5
	contributing	5	2.5	2.5	100.0
	Total	199	100.0	100.0	

website maintainance

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	not contributing	193	97.0	97.0	97.0
	contributing	6	3.0	3.0	100.0
	Total	199	100.0	100.0	

Documentation

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	not contributing	190	95.5	95.5	95.5
	contributing	9	4.5	4.5	100.0
	Total	199	100.0	100.0	

Translations

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	not contributing	155	77.9	77.9	77.9
	contributing	44	22.1	22.1	100.0
	Total	199	100.0	100.0	

Board work/accounts maintainance

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	not contributing	179	89.9	89.9	89.9
	contributing	20	10.1	10.1	100.0
	Total	199	100.0	100.0	

promotional and advocacy

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	not contributing	175	87.9	87.9	87.9
	contributing	24	12.1	12.1	100.0
	Total	199	100.0	100.0	

artwork

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	not contributing	196	98.5	98.5	98.5
	contributing	3	1.5	1.5	100.0
	Total	199	100.0	100.0	

Long-term view of Contribution:GNOME Foundation respondents

	Paid to work on F/OS	Paid to work only on GNOME	Paid to work on GNOME and F/OS	Paid to work on GNOME and GNOME & F/OS	All Paid
Volunteers	11: x2=0.38, N=126, F=0.7777	34: x2<0.001, N=136,	38: x2<0.001, N=133, F=13.475	12: x2<0.001, N=171, F=42.384	10: x2<0.001, N=199,
Paid to work on F/OS		15: x2<0.001 , N=66, F=27.981	14: x2=0.003, N=63, F=9.544	13: x2<0.0013, N=73, F=17.334	
Paid to work only on GNOME			16: x2=0.003, N=73, F=9.788		

Short-term involvement: Main Area of Contribution: GNOME Foundation respondents

	Paid to work on F/OS	Paid to work only on GNOME	Paid to work on GNOME and F/OS	Paid to work on GNOME and GNOME & F/OS	All Paid
Volunteers	3 ANOVA	33a, 33 ANOVA 33a: χ^2=(4, N=136) 33 ANOVA: core: x2<0001, main:x2=0.037, sec:x2=0.004, dev:x2=0.684, peract:x2=0.023	37 ANOVA 37 ANOVA: core: x2<0.001, main: x2=0.022, sec: x2=0.003, dev:x2=0.061, peract:x2=0.017	4a, 4b, 4ANOVA 4a: x2<0.001 (4, N=171) 4b: (1, N=171)core:x2<0.001, main: x2=0.01, sec: x2<0.001, dev.: x2=0.338, peract:x2=0.003 ANOVA confirms the above	2a, 2b, 2ANOVA 2a: x2<0.001 (4, N=199) 2b:(1, N=199)core:x2 <0.001, main: x2=0.018, sec: x2<0.004, dev.: x2=0.556, peract:x2=0.01 9 ANOVA confirms the above
Paid to work on F/OS		7a, 7b, 7ANOVA 7a: x2<0.001 (4, N=66) 7b: (1, N=66) core: x2<0.001, main:x2=0.481, sec:x2=0.015 dev:x2=0.909 peract:x2=0.062	6a, 6ANOVA 6a: x2=0.001 (4, N=63) 6ANOVA: core: x2=0.003, main:x2=0.397, sec:x2=0.011 dev:x2=0.093 peract:x2=0.048	5a, 5b, 5ANOVA 5a: x2<0.001 (4, N=101) 5b: (1, N=101) core: x2=0.001, main:x2=0.389, sec:x2=0.003 dev:x2=0.287 peract:x2=0.022 ANOVA confirms the above	
Paid to work only on GNOME			8a, 8b, 8ANOVA 8a: x2=0.0428 (4, N=73) 8b: (1, N=73) core: x2=0.211, main:x2=0.852, sec:x2=0.858 dev:x2=0.074 peract:x2=0.858		

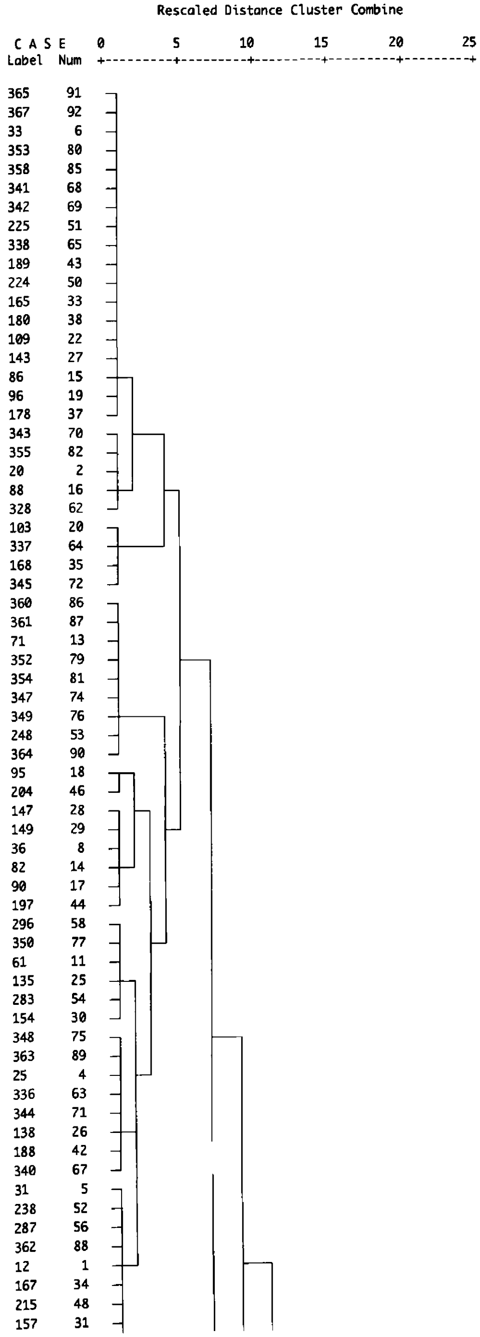
GUADEC Attendance: Detailed overview of differences between Affiliated and Volunteers

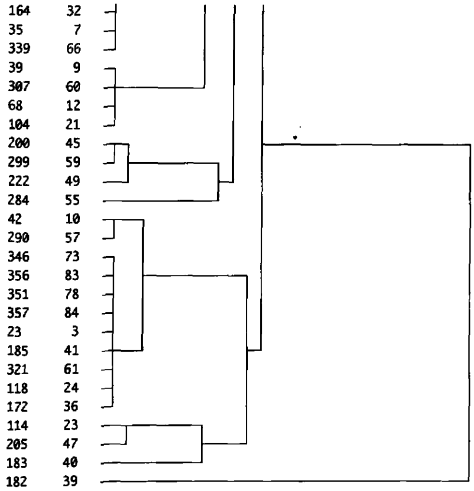
	Paid to work on F/OS	Paid to work only on GNOME	Paid to work on GNOME and F/OS	Paid to work on GNOME and GNOME & F/OS	All Paid
Volunteers	19ANOVA: x2=0.2, N=126	35: x2<0.001 (3, N=136)	39: x2<0.001 (3, N=137)	20: x2<0.001 (3, N=170)	18: x2<0.001 (3, N=198)
Paid to work on F/OS		23: x2=0.001 (3, N=62)	22: x2=0.12 (3, N=66)	21: x2<0.001 (3, N=100)	
Paid to work only on GNOME			24: x2=0.523 (3, N=72)		

GNOME Maintainers Dendrogram

* * * * * * H I E R A R C H I C A L   C L U S T E R   A N A L Y S I S * * * * * *

Dendrogram using Complete Linkage




164 32
35 7
339 66
39 9
307 60
68 12
104 21
200 45
299 59
222 49
284 55
42 10
290 57
346 73
356 83
351 78
357 84
23 3
185 41
321 61
118 24
172 36
114 23
205 47
183 40
182 39

ANOVA GNOME exclusive maintainership*Main Area Modules

		N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for		Minimum	Maximum
						Lower Bound	Upper Bound		
CPLATFOR	maintained exclusively by non affiliated	36	0.25	0.43916	0.07319	0.1014	0.3986	0	1
	maintained exclusively by affiliated	67	0.4478	0.50102	0.06121	0.3256	0.57	0	1
	Total	103	0.3786	0.48742	0.04803	0.2834	0.4739	0	1
CMAINDES	maintained exclusively by non affiliated	36	0.1111	0.31873	0.05312	0.0033	0.219	0	1
	maintained exclusively by affiliated	67	0.2388	0.42957	0.05248	0.134	0.3436	0	1
	Total	103	0.1942	0.3975	0.03917	0.1165	0.2719	0	1
CSECDES	maintained exclusively by non affiliated	36	0.25	0.43916	0.07319	0.1014	0.3986	0	1
	maintained exclusively by affiliated	67	0.1642	0.37323	0.0456	0.0731	0.2552	0	1
	Total	103	0.1942	0.3975	0.03917	0.1165	0.2719	0	1
CAPPLICA	maintained exclusively by non affiliated	36	0.1944	0.40139	0.0669	0.0586	0.3303	0	1
	maintained exclusively by affiliated	67	0.1045	0.30819	0.03765	0.0293	0.1797	0	1
	Total	103	0.1359	0.34438	0.03393	0.0686	0.2032	0	1
COTHER	maintained exclusively by non affiliated	36	0.1944	0.40139	0.0669	0.0586	0.3303	0	1
	maintained exclusively by affiliated	67	0.0448	0.20837	0.02546	-0.0061	0.0956	0	1
	Total	103	0.0971	0.29752	0.02932	0.0389	0.1552	0	1

ANOVA GNOME exclusive maintainership*Main Area Modules

		Sum of Squares	df	Mean Square	F	Sig.
CPLATFOR	Between Groups	0.916	1	0.916	3.967	0.049
	Within Groups	23.317	101	0.231		
	Total	24.233	102			
CMAINDES	Between Groups	0.382	1	0.382	2.451	0.121
	Within Groups	15.735	101	0.156		
	Total	16.117	102			
CSECDES	Between Groups	0.172	1	0.172	1.093	0.298
	Within Groups	15.944	101	0.158		
	Total	16.117	102			
CAPPLICA	Between Groups	0.19	1	0.19	1.608	0.208
	Within Groups	11.908	101	0.118		
	Total	12.097	102			
COTHER	Between Groups	0.525	1	0.525	6.23	0.014
	Within Groups	8.505	101	0.084		
	Total	9.029	102			

Means Analysis of GNOME maintainers according to types and number of modules maintained

		no of component model modules maintained	no of graphical libraries modules maintained	no of accessibility libraries modules maintained	no of other core platform modules maintained	no of core desktop components maintained	no of Evolution modules maintained	no of Multimedia modules maintained	no of secondary desktop libraries maintained	no of secondary desktop components maintained	no of user apps maintained	no of developer tool modules maintained	no of peripheral modules maintained
1	Mean	0.01	0.16	0.03	0.13	0.12	0.15	0.03	0.09	0.12	0.23	0.03	0.07
	N	75	75	75	75	75	75	75	75	75	75	75	75
	Std. Deviation	0.115	0.436	0.231	0.414	0.327	0.392	0.162	0.293	0.327	0.421	0.162	0.251
2	Mean	0.5	1.17	0.67	0.83	0	0	0	0.17	0.17	0.17	0	0
	N	6	6	6	6	6	6	6	6	6	6	6	6
	Std. Deviation	1.225	1.472	1.211	1.329	0	0	0	0.408	0.408	0.408	0	0
3	Mean	0	0	0	0	0.17	0	0	0.83	0.33	0.17	0.83	0.17
	N	6	6	6	6	6	6	6	6	6	6	6	6
	Std. Deviation	0	0	0	0	0.408	0	0	0.983	0.516	0.408	0.983	0.408
4	Mean	3	0	0	1	6	0	0	0	1	0	0	0
	N	1	1	1	1	1	1	1	1	1	1	1	1
	Std. Deviation	.	.	.	.	.	.	.	.	.	.	.	.
5	Mean	0	0	0	0.5	2.5	0	0.75	0	0	0	0	0
	N	4	4	4	4	4	4	4	4	4	4	4	4
	Std. Deviation	0	0	0	1	1	0	1.5	0	0	0	0	0
Total	Mean	0.08	0.21	0.07	0.2	0.28	0.12	0.05	0.14	0.14	0.21	0.08	0.07
	N	92	92	92	92	92	92	92	92	92	92	92	92
	Std. Deviation	0.45	0.584	0.387	0.559	0.856	0.358	0.343	0.408	0.35	0.407	0.339	0.248

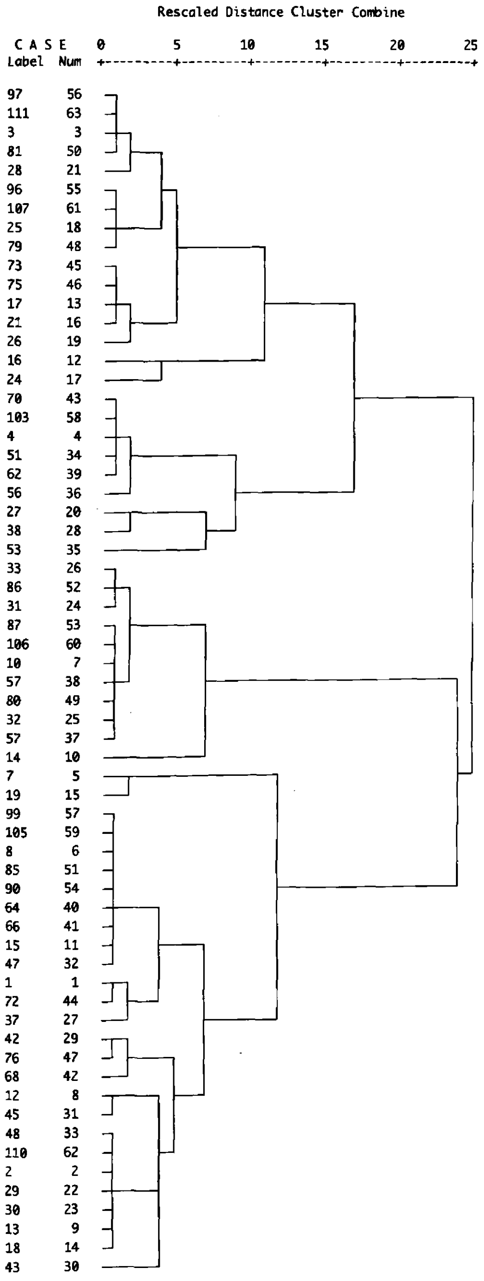
Organizations sponsoring KDE e.V. respondents

no	Organization	Paid to contribute only to KDE	Paid to contribute to KDE and F/OS	Paid to contribute to F/OS	Total
1	Subcontracted/Other	2	1	1	4
2	Re/Source Inc.	1			1
3	Linspire, Inc.	1			1
4	Linux4MEDIA GmbH		1		1
5	Linupfront GmbH			1	1
6	Mandriva Labs-Brazil		1		1
7	Basyskom GmbH	1			1
[illegible]	Red Hat, Inc.		1		1
9	University of Nijmegen			1	1
10	Sirius Corporation	1			1
11	Novell, Inc./SUSE	4	1		5
12	Klarälvdalens Datako:	3			3
13	Trolltech ASA	2	1		3
14	Staikos Computer Services, Inc.			1	1
15	Intel Corporation	1			1
16	Froglogic GmbH			1	1
17	Credativ GmbH	1			1
18	Trolltech ASA & Klar:	1			1
	TOTAL	18	6	5	29

KDE e.V. RESPONDENTS DENDROGRAM

******HIERARCHICAL CLUSTER ANALYSIS******

Dendrogram using Complete Linkage



Organizations sponsoring KDE maintainers

	Name of Company	No of Developers
1	Subcontracted/Other	4
2	Trolltech ASA	5
3	Trolltech ASA and Klarälvdalens Datakonsult AB	1
4	Klarälvdalens Datakonsult AB	5
5	Froglogic GmbH	2
6	SAP Linux Lab	2
7	University of Nijmegen	1
8	Novell, Inc	8
9	Staikos Computer Services, Inc.	1
10	SourceXtreme, Inc.	1
11	Hubcapconsulting	1
12	Caldera Systems, Inc.	1
13	Credativ, GmbH	1
14	Intel Corporation	1
15	Open Office Polska LLC	1
16	Mandrake, Inc.	1
17	Conectiva/Mandriva, Inc.	1
18	Duskglow Consulting LLC	1
19	Sirius Corporation	1
	Total	39