Paper to be presented at the DRUID Academy 2013

on

DRUID Academy 2013
Comwell Rebild Bakker, Rebild/Aalborg

Linux Contribution and Patents, A Firm Level Analysis

Altay Özaygen
Institut Mines-Télécom, Télécom Ecole de Management
Management, Marketing and Strategy Department
altay.ozaygen@telecom-em.eu

Abstract
This empirical work provides a quantitative study of patenting behavior among firms contributing to the free and open source software project the Linux kernel. This study collects a data set of 169 firms among 800 firms which have contributed to the Linux kernel for seven years period (2005-2011). The analyzed 169 firms contributed to the Linux kernel with 48.7 % of the total changeset during these seven years. We show that patenting and contributing to the Linux kernel project have a positive interaction.

Jelcodes:L20,-

Linux Kernel Contribution and Patents, A Firm Level Analysis

Altay Özaygen*

Institut Mines-Télécom, Télécom Ecole de Management

Abstract

This empirical work provides a quantitative study of patenting behavior among firms contributing to the free and open source software project the Linux kernel. This study collects a data set of 169 firms among 800 firms which have contributed to the Linux kernel for seven years period (2005-2011). The analyzed 169 firms contributed to the Linux kernel with 48.7 % of the total changeset during these seven years. We show that patenting and contributing to the Linux kernel project have a positive interaction.

Keywords: panel data, Linux, software patents, free and open source software (FOSS)

1 Introduction

The innovation model and the regimes of appropriation behind free and open source software (FOSS) development differs from that of the proprietary software domain. FOSS development has traditionally been associated to hacker ethic which encompass values such as access to computers, free information, mistrust to the authority and promotion of decentralization, evaluations based only on hacking capabilities, search of art and beauty on computers and better life through computing (Levy, 2010, chapter 2). Since the 1980s FOSS development and usage gained an important momentum. At the same time a patent explosion is observed. This increase is

*altay.ozaygen@telecom-em.eu

attributed to various facts such as the electrical, electronics, computing, and scientific instruments industries research performances (Hall, 2004), to the innovation of management (Kortum and Lerner, 1999), to the investment of the universities and public research institutes into patenting (Mowery, 2001) or to the strategic patenting of firms (Shapiro, 2001; Bessen and Hunt, 2007).

There is also an important debate on the necessity of software patents. It is suggested that a tight intellectual property right (IPR) regime would increase the number of patent litigation (Dosi et al., 2006) which is subsequently demonstrated for software patents. It is shown that 94% of the patent lawsuits are related to software patents (Allison et al., 2009). Within this legal environment, many FOSS advocates and software developers claim that software patents became an important threat to the development of FOSS. Moreover, the existing literature offers different arguments in opposition to the software patents in general (Nelson, 1994; Bessen, 2009; Bessen et al., 2011).

Contrary to the FOSS advocates, there is not a clear cut limits between firms working in FOSS and proprietary software domain. Many projects, which are sponsored by or continue to be developed within firms, were either an academic or a voluntary based FOSS projects. Some of these firms have also a patent portfolio which sometimes also contains software patents. Moreover, firms, which have started as an example of a FOSS based company, are also filing various patents. In previous researches, it is demonstrated that firms entering into the FOSS field, adopt a hybrid business model which is a combination of proprietary and FOSS product development under different licensing schemes (Bonaccorsi et al., 2006; Harison and Koski, 2010). It is also shown that 75% of the Linux kernel, one of the successful flagship of the FOSS movement is developed by those being paid for their work in the project (Corbet et al., 2012). These results show that at least in the case of Linux kernel, which was at its debut a student project, became mainly a corporate product without excluding the hacker ethic due to its restrictive licensing scheme. The Linux kernel is licensed with the General Public License (GPL) which protects any appropriations of the code (Stallman, 2002). The question arises whether contribution to FOSS and patenting have a positive interaction and not a contradictory activities among firms contributing to the Linux kernel.

This work aims to contribute to the literature by examining patenting and FOSS development activities on firms' performance through econometric evidences. This empirical work is limited to the development of the Linux kernel. Related to these observations on FOSS and software patents, this research will focus on the following research questions; How contribution to the Linux kernel project and patenting contributes to the firm performance and are these two activities contradictory?

Three databases are used; from PATSTAT database; patent information, from Thomson One Banker; firm level data such as R&D expenditure, number of employees, sales etc. and from Linux Foundation, kernel contributors data are obtained. Using panel data econometrics a set of 169 firms among 800 firms and other entities such as volunteers, public research institutes etc. which have contributed to the Linux kernel for seven years period (2005-2011) are analyzed. Contribution to the Linux kernel is measured by the number of the changeset i.e. the group of modification containing files which are relevant to each other. The analyzed 169 firms contributed to the Linux kernel with 48.7 % of the total changeset during these seven years.

In this research we show that the general patenting and contributing to the Linux kernel project affect positively firms' sales, moreover, their interaction also has a positive effect on firms' performance.

The next section deals with the literature on FOSS, Linux kernel development and software patents and proposes our hypotheses. Section 3 gives and overview of the data, Section 4 shows the results of the analysis and discusses results.

2 Patents, FOSS and hypotheses

A patent gives a monopoly right to the inventor by excluding others to use a particular invention for a limited time. Patents aim to encourage economic agents to innovate (Hall, 2007). However, effect of patents are not the same for all technologies. Even though there are various views in this domain, patents are accepted to promote innovation in chemical, pharmaceutical and biotechnology industry (Arora et al., 2001). However, it is argued that strong patent protection has a deterrent effect on innovative activities in cumulative technologies such as software, radio and aircraft technologies (Mazzoleni and Nelson, 1998; Nelson, 1994). It is also suggested that strong IPR regime did not play an important role in the emergence of the ICT and it is not a tool for value generation. On the contrary weak IP regimes played an important role in the emergence and diffusion of transistor, semiconductor and mobile telephony (Dosi et al., 2006).

The USPTO accepts software patents since 1981 but it has been suggested that software industry has a weak patent protection (Bessen, 2009). On the other hand, even though European Patent Office rejects patents related to computer programs as it is, the European patent legislation does not prevent to grant any software related patents and there is a large number of patenting activities in this field (Rentocchini, 2011).

According to one research stream patenting does not eliminate competition but it increases the cost of imitation (Mansfield et al., 1981). It is also argued that a strong intellectual property right (IPR) regime creates hurdles to the following researches (Dosi et al., 2006; Bessen and Hunt, 2007). Strategies like patent thicketing (Shapiro, 2001) within a "cumulative system technology" (Nelson, 1994) in a strong IPR regime creates lots of legal complexities in industries such as software. Patent lawyers and non-practicing entities (NPE or patent troll) become important figures in the software industry due to the increasing number of patent litigation cases. It is ironically pointed out that the number of patent lawyers is growing faster than the amount of research (Barthon, 2000). It is reported that 94% of the patent lawsuits are related to software patents (Allison et al., 2009). NPEs, firms that litigate over software patents with unpredictable boundaries have caused an important financial burden to the defendant firms. It is calculated that there is a half a trillion dollar lost of wealth from 1990 through 2010 for the defendant technology firms (Bessen et al., 2011). According to the authors, that loss harms society and there is little evidence of money transfer to independent inventors from non producing entities' litigation.

The property rights defended by the FOSS movement is fundamentally in the opposite side of the patent rights. FOSS property scheme is "around the right to distribute, not the right to exclude" (Weber, 2004). The licensing schemes of the FOSS ensure that the source code is kept in the public domain. Among various licensing scheme, the GPL is one of the most popular one within the FOSS world. GPL license family (GPLv2, GPLv3, LGPLv2, and LGPv3) usage is about 54% of all software packages released under FOSS license (BlackDuckSoftware.com, 2012). The GPL is developed by the Free Software Foundation which also provides various software, tools to the FOSS world since 1984.

The GPL is in its third version and keeps evolving with the new development of the software industry. The GPL gives roughly four rights to the owner of the code; 1) the right to run for any purposes, 2) the right to read the source code, 3) the right to change it and 4) the right to give the code to anyone. However, any redistribution of the software whether changed or not should be done with the source code. GPL is not compatible with patents. It contains a clause which does not allow the development of a software requiring licensing due to patents. However, the BSD license which is another popular licensing scheme in the FOSS realm do not have any specific clause on patents. The Linux kernel adopted GPL since its debut and is developed with the second version.

In this research the patenting behaviors of firms contributing to the Linux kernel is analyzed. Linux kernel is the operating system kernel which powers over 90% of the fastest 500 computers in the world in 2012 but it is also used in PCs, laptops, smart-phones, routers and many other electronic appliances. It is the main link between the hardware and the applications software. In order to run a computer with FOSS some other software are also needed. A set of FOSS software with the Linux operating system is called a GNU/Linux distribution. It is calculated that the number of lines of the Linux kernel code represent only 6% percent of a whole set of a desktop oriented Linux distribution (Amor et al., 2005). The Linux kernel project started in 1991 targeting Intel x86 based PCs. It is considered as one of the most successful, a show case among FOSS projects.

Many manufacturers contribute to the development of the Linux kernel project. They put this software at the core of their hardware. On the other hand, some other hardware manufacturers could refrain to contribute to a project such as Linux in order not to reveal the internal mechanism of their hardware. They could also prefer not to invest an in-house FOSS development for their hardware. In that case, the community could develop FOSS code in order to run Linux on top of the hardware which misses Linux support. This activity requires reverse engineering and sometimes the result is not on par if the hardware company could have given at least the blue prints about their hardware. There are many cases in which firms reveal their code selectively. The main reasons of this openness are, the enforcement of the GPL, the positive image of the company within the community and the reduction of the cost of maintenance of the revealed code (Henkel, 2006).

There are firms which base their business model on hybrid models (Bonaccorsi et al., 2006; Dahlander and Magnusson, 2005). The hybrid business model consists of revenues generated by the combination of the traditional software licensing fees and FOSS related services. The hybrid business model is used for an entry strategy. It is shown that the hybrid business model is not transitional i.e. the firm which opts a hybrid model after its entry do not change its business model into a pure FOSS development or into a pure proprietary software development (Bonaccorsi et al., 2006). Among Italian software firms it is observed that there are lock-in effects for switching to FOSS on both the supply and the demand side. Firms producing FOSS products and services are driven mainly by economic and technological factors rather than the social factors (Bonaccorsi and Rossi, 2003). The coexistence of proprietary and FOSS with different remuneration modes among software developers are interpreted as the key to the success of the FOSS movement.

Patenting and contributing to FOSS project could seem contradictory. Depending on the industry, patenting and contributing to a FOSS project could be the result of a varying patenting and innovation strategies. It is reported that software patents are used strategically by established firms to build thickets to restrain competition (Bessen and Hunt, 2007). Another strategy is to add market value to the company through patenting for a possible acquisition. When the value of software patents is questioned, it is not possible to find a clear answer if those patents increase the value of a pure software firm (Hall and MacGarvie, 2010). However, patenting have a positive effect on the IPO values of software firms in European and US market (Useche, 2012).

There are three parties within the FOSS community who argues that software patents are inherently an important threat to FOSS. The first party are the developers; it is very unlikely that a FOSS project members check whether the project constituents are infringing any patents. Most of the time a FOSS project start because few developers feel the need to develop it due to a "scratching the personal itch" (Raymond, 1999). Independent developers, in case of an infringement suit, are at the mercy of the patentee and most of the FOSS developers do not have any sufficient resources to afford a legal dispute against a patent defendant. The second group who sees the software patents as an important threat are made of corporate FOSS users and developers. Within this second group there are many Fortune 500 firms which represent the most wealthy prey for patent trolls. And the last group are the end users who also feel the threat of patents which could damage the development of their favorite software.

A strategy to protect the development of the Linux project is the creation of a patent portfolio which would be licensed to entities which in turn agree not to use their own patent to sue any Linux and Linux related software developers. This patent portfolio is created by the Open Innovation Network[1], with was founded in 2005 by IBM, Novell, Philips, RedHat and Sony. This company holds more than 600 US patents which are either given by supporting firms or are acquired.

This paper aims to understand if the contribution to the Linux kernel project and patenting have an effect on the firm's performance criteria. According to the literature review given above we test the following hypothesis;

[1]http://www.openinventionnetwork.com/about.php

H1: Patenting activities within the Linux contributing firms affects positively to the firms' performance.

H2: The interaction of patenting and contribution to Linux affects positively to the firms' performance.

3 Data and econometric models

The data on the contribution to Linux kernel consists of number of changeset given by firms during seven years, from the start of 2005 to the end of the year 2011. During this seven years there are 32 Linux releases, one release for every 8 to 12 weeks are observed. During this time period almost 800 different firms have contributed. Figure 1 shows top contributor firms with aggregated values for "Consultants", "Public institutes", "Unknown" and "None" groups. "Public institutes" comprises universities, research institutes and contribution made from developers working in some ministries. "Unknown" represents contributors origin which are not found and "None" represents developers who are known to contribute without any salary from any firm. Figure 2 shows aggregated analyzed contributor firms with respect to firms which are not analyzed and all other entities such as volunteers, consultants, public institutes etc.

For this research we have obtained data for 360 firms from Thomson One Banker but only 169 of them comprises data on firms characteristics. Currency change have been carried out through OECD data and for three non-OECD countries oanda.com is used. To check the correctness of the processed firm level data ychart.com is also used but in case of differences, data provided by Thomson One Banker is taken into account.

There is an increase of nearly 3 folds from 2005 to 2011 in the number of Linux contributing firms. The number of firms analyzed in this study is 169 and among 35 of them do not have any patent application, and 41 of them do not have any published patent between 2005 and 2011. Figure 3 shows firms' published patent distribution.

In this research the number of contribution to the Linux kernel project is evaluated by the number of changeset. A changeset is a group of modification on Linux kernel containing files which are relevant to each other. The distribution of the number of changeset among firms are given in Figure 4. The number of changeset is summed up for each entity so that the number of changeset contribution per year

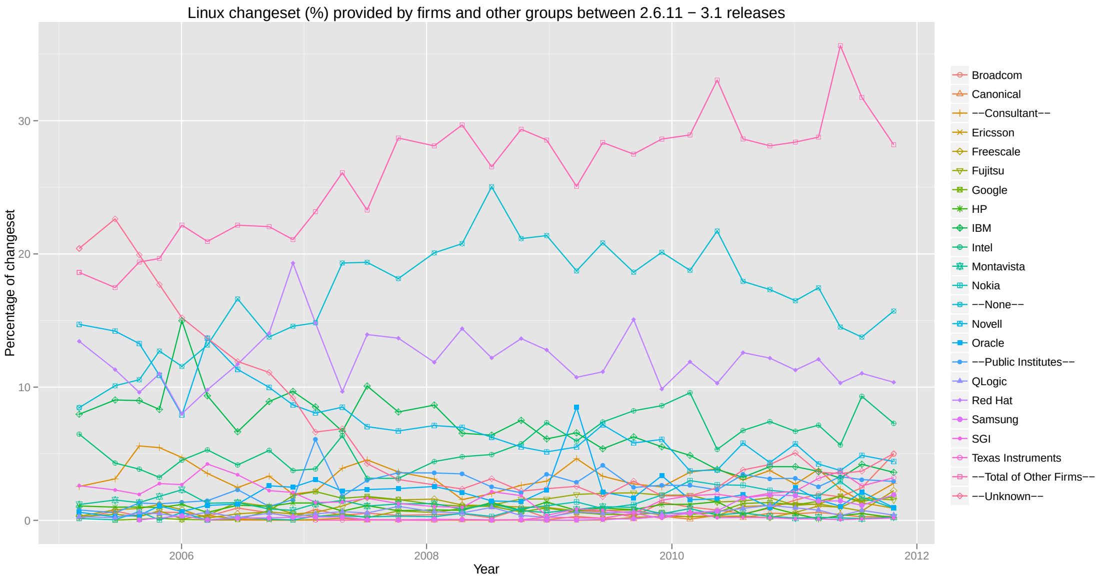


Figure 1: Top contributor firms with respect to “Consultants”, “Public institutes”, “Unknown” and “None”.

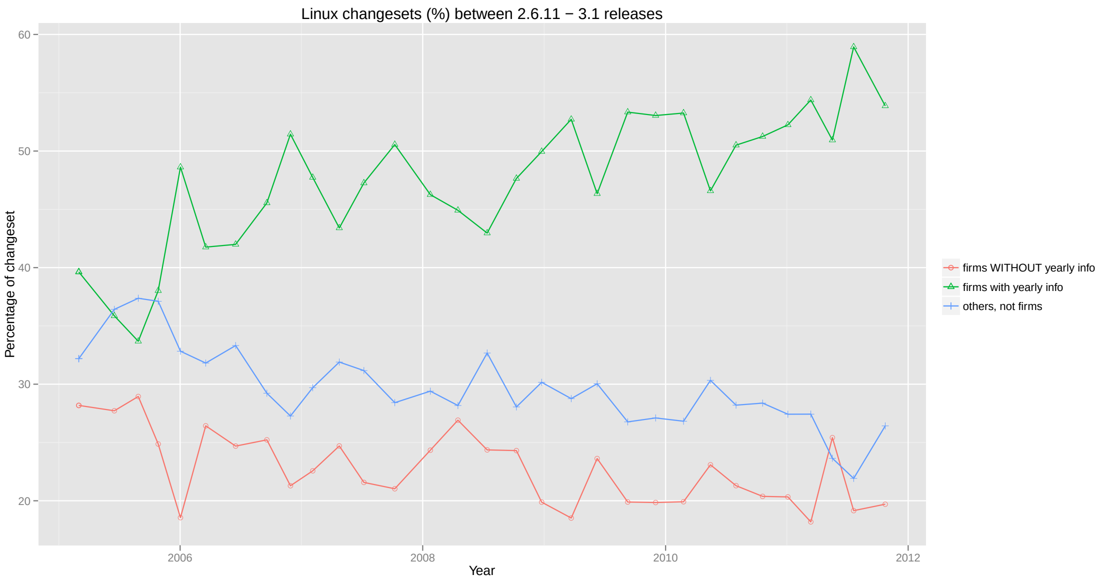
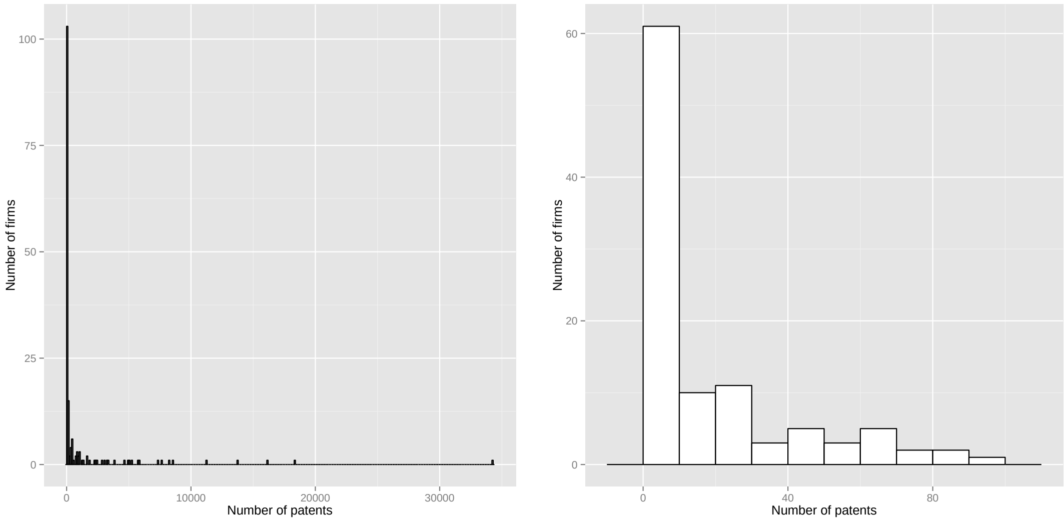


Figure 2: Analyzed contributor firms with respect to firms without data and entities such as universities, research institutes and voluntaries which do not have any entries in business databases.

of an entity could be obtained.

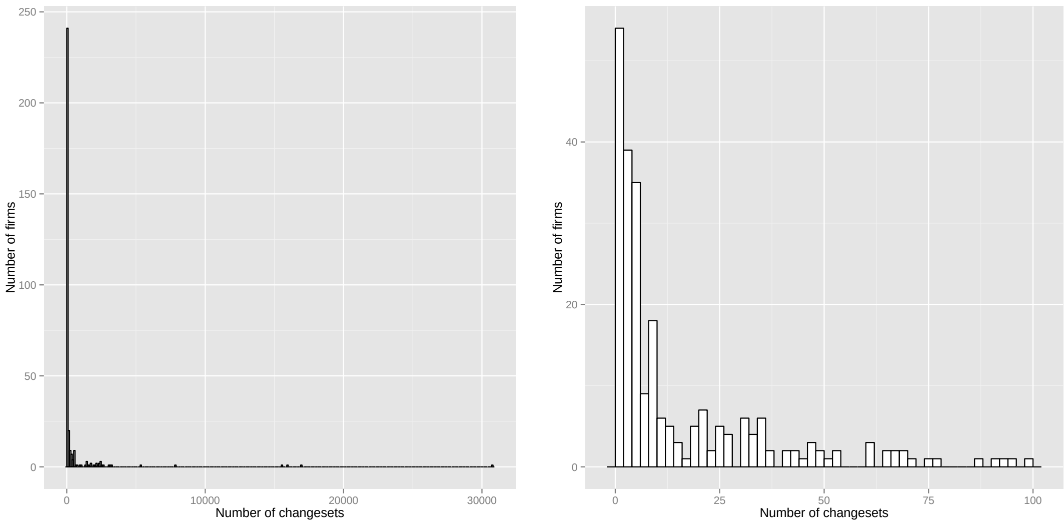
In this research we use PATSTAT 2012 patent database and for the patent statis-



(a) All firms

(b) Firms with < 100 patents

Figure 3: Number of patents and number of firms.



(a) All firms

(b) Firms with < 100 changesets

Figure 4: Number of changesets and number of firms.

tics we only considered patent applications to the USPTO. The definition of a software patent is unclear, it does not have a distinct IPC code which make it difficult to evaluate statistically the number of software patents. In this article the number of software patents detained by each firm is obtained through the method used by Hall and MacGarvie (2010). In this method software patents are obtained by the intersection of two sets of patents which are constructed through the definitions of software patents given in several previous researches. The first set is obtained by the definitions of Mowery and Graham (as cited in Hall and MacGarvie (2010)) and Hall and MacGarvie (2010). The second set is obtained by the search criteria given by Bessen (2010). The first set is the sum of the patents in “Electric Digital Data Processing” (G06F), “Recognition of Data; Presentation of Data; Record Carriers; Handling Record Carriers” (G06K) and “Electric Communication Technique” (H04L) IPC classes (Mowery and Graham) and all patents in IPC classes of patents awarded to the fifteen of the largest US software firms given by Hall and MacGarvie (2010). The second group is made of patents obtained by the method used by Bessen (2010) which is adapted to the PATSTAT database. This method consists of keyword search on title and abstract. Keywords used are “software” or the words “computer” and “program” in abstract but these patents should not contain “semiconductor”, “chip”, “circuit”, “circuitry” or “bus” in their title. Patents containing “antigen”, “antigenic”, or “chromatography” in patent abstracts are also excluded. From this set of software patent applications, we have counted patent applications to the firms active in the development of the Linux kernel.

rd_expense	R&D expenses.
sales	sales in USD.
total_assets	the sum of total current assets, long term receivables, investment in unconsolidated subsidiaries, other investments, net property plant and equipment and other assets.
changeset	group of modification on Linux kernel containing files which are relevant to each other.
changeset_stock	stock number of changeset since 2002, with 15% depreciation.
rd_stock	stock value of R&D expenses calculated starting 2002, with 15% depreciation rate.
(soft_)pat_stock	(software) patent stock calculated starting 1990, with 15% depreciation rate.

Table 1: Variable definitions.

The number of patents and contribution to Linux will vary across industries and time due to differences in the factors affecting the decision to patent and contribute to Linux. The table 2 gives the distribution of firms analyzed in this paper according to their Global Industry Classification Standard (GICS) code. The group named “other” is the aggregation of industries such as transport, media, capital goods, automotive, health etc. There are also 18 firms which do not have any GICS code.

Hardware	56
Semiconductor	44
Software	31
Other	38

Table 2: Industry distribution according to the GICS codes of the analyzed 169 firms.

	N	mean	sd	min	max
rd_expense(*)	1047	874.64	1628.12	0.00	9043.00
sales(*)	1119	12879.02	28176.61	0.23	206395.59
total_assets(*)	1113	18531.07	65926.94	0.96	797800.00
changeset	1121	109.77	475.70	0.50	6748.00
yearly_appln_pat_num	1121	263.89	719.32	0.50	9378.00
yearly_appln_soft_pat_num	1121	13.51	75.36	0.50	1536.00
rd_stock	1052	3340.97	6543.44	0.15	41700.99
pat_stock	1121	926.85	2476.57	0.23	24300.00
soft_pat_stock	1121	20.83	122.42	0.50	2600.00
linux_stock	1121	281.40	1387.02	0.50	21300.00
log(sales_over_assets)	1050	14.95	1.38	8.43	24.22
log(rd_stock_over_assets)	1045	-1.34	1.29	-9.99	1.71
log(pat_stock_over_assets)	1113	-3.71	2.28	-11.56	0.67
log(soft_pat_stock_over_assets)	1113	-6.73	1.98	-13.12	-0.65
log(changeset_stock_over_assets)	1052	-4.34	2.89	-11.02	4.58

Table 3: Descriptive statistics (* in millions).

Table 4: Correalation matrix.

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
rd_expense_usd	1	1														
sales_usd	2	0.8	1													
total_assets_usd	3	0.519	0.766	1												
changeset	4	0.248	0.124	0.052	1											
yearly_appln_pat_num	5	0.641	0.572	0.362	0.239	1										
yearly_appln_soft_pat_num	6	0.433	0.314	0.153	0.344	0.762	1									
rd_stock	7	0.973	0.776	0.504	0.257	0.582	0.399	1								
pat_stock	8	0.679	0.627	0.385	0.332	0.763	0.575	0.702	1							
soft_pat_stock	9	0.439	0.335	0.172	0.299	0.386	0.491	0.496	0.663	1						
linux_stock	10	0.23	0.127	0.055	0.929	0.194	0.294	0.251	0.34	0.407	1					
log(sales_over_assets)	11	0.022	0.259	0.152	-0.055	0.138	0.012	-0.026	0.096	-0.01	-0.051	1				
log(rd_stock_over_assets)	12	-0.041	-0.273	-0.246	0.011	-0.145	-0.01	0.016	-0.092	0.012	0.014	-0.918	1			
log(pat_stock_over_assets)	13	-0.007	-0.064	-0.085	0.064	0.161	0.09	0.024	0.255	0.111	0.073	-0.371	0.412	1		
log(soft_pat_stock_over_assets)	14	-0.268	-0.381	-0.3	0.039	-0.108	0.034	-0.211	0.006	0.116	0.063	-0.359	0.403	0.551	1	
log(changeset_stock_over_assets)	15	-0.24	-0.267	-0.239	0.346	-0.138	-0.007	-0.222	-0.055	0.031	0.317	0.059	-0.071	0.066	0.402	1

3.1 Model

In this paper we use the sales volume as the firm performance criteria and we are evaluating the firms' tangible assets and intangible assets effects on sales volume. Tangible (physical) assets are those which are recorded and which could be used in calculations. However, intangible ones are not easy to count. They are the byproduct of the R&D activities, and in many cases patents and other patent features are used as the main proxy for the evaluation of the firms' intangible assets.

The model representing the assets of the firm i in year t with respect to its sales value is:

$$S_{it} = q_t(A_{it} + gK_{it}) \tag{1}$$

where S is the sales value in USD of the firm i, A is the sum of the tangible assets such as plant, equipment and financial assets, K is the intangible "stock of knowledge" which is the sum of the past R&D. The number of patents and other patent related attributes like citation numbers are generally used as a proxy to represent the firms stock of knowledge. The manipulation of this model results with the log of sales over assets value as a function of the knowledge asset intensity.

$$log\frac{S_{it}}{A_{it}} = f(A_{it}, K_{it}) \tag{2}$$

As the knowledge assets cannot be measured directly we are using the R&D expenses, number of patents and number of contribution to the Linux kernel to represent the knowledge assets of firms in our data set. The function representing the knowledge asset K of firm i at year t is given in equation 3.

$$K_{it} = g[RD_{it}, P_{it}, L_{it}] \tag{3}$$

where R_{it} is the R&D stock, P is the patent stock and L is the Linux knowledge stock in year t of the firm i. In this paper we use the R&D expenditure in USD or converted to USD for non-US firms. The number of patent stock is obtained by the number of patent application and the number of Linux kernel contribution is given by the number of changesets. The number of patent application is used to reflect the yearly number of the "patentable" innovation of the same year.

Due to the heterogeneity and skewness of the sales, patenting and contribution to the Linux kernel, we use a simple first order logarithmic equation of the log of sales over assets given in equation 4.

$$log\frac{S_{it}}{A_{it}} = logq_t + \gamma_0 logA_{it} + \gamma_1 log\frac{RD_{it}}{A_{it}} + \gamma_2 log\frac{P_{it}}{A_{it}} + \gamma_3 log\frac{L_{it}}{A_{it}} + \epsilon_{it} \quad (4)$$

The knowledge assets are not used as it is. A firm's stock of knowledge is increased by the investment done through yearly R&D expenditure, number of patents obtained etc. but there is also some depreciation of the older knowledge investment with time (Hall et al., 1986). The knowledge stock is calculated using the usual 15% of depreciation rate over years. In equation 5 an example of the knowledge stock with depreciation depending on the R&D spending is shown.

$$K_{it}^R = (1-\rho)K_{i,t-1}^R + R_{it} \quad (5)$$

where R_{it} is the R&D spending for the year t for the firm i and the ρ is the 15% depreciation rate. K_{it}^R is the stock of knowledge which increases every year with the investments but there is also a depreciation of this stock. Similar calculations are done for the knowledge stock depending on patent application and on Linux contribution. The R&D stock is calculated starting from year 2002, the patent stock is obtained from data starting from 1990 and the knowledge stock on Linux starts from 2005. If the firm is established later than these dates then the calculations are done based on the firms' establishment year.

4 Results and discussion

In this analysis we have used data of the 169 firms which are available on Thomson One Banker. These firms have contributed in total to the 48.7% of the submitted changeset to the Linux kernel project during the seven years period. Results of the fixed effect panel data of the equation 4 is given in Table 5. The results of the regression analysis show that the stock of the patent and the stock of the Linux contribution (changeset) contribute positively to the sales value. The positive interaction of patenting and contribution to the Linux kernel project on sales performace shows that producers are while reveling some of their knowledge are also patenting some other portions of their knowledge. From the results, we denote that the sales volume is not affected by the number of software patents.

We can discuss our results according to the two findings within the FOSS developing firms; the first one is the hybrid business model (Bonaccorsi et al., 2006; Dahlander and McKelvey, 2005) and the second one is related to the IP modularity

	Model 1	Model 2	Model 3
log(total_assets)	−0.12	−0.12	−0.13
	(0.09)	(0.09)	(0.09)
log(rd_stock_over_assets)	−0.82***	−0.82***	−0.84***
	(0.10)	(0.10)	(0.10)
log(soft_pat_stock_over_assets)		−0.03	−0.03
		(0.02)	(0.02)
log(pat_stock_over_assets)		0.04**	0.08***
		(0.02)	(0.03)
log(linux_stock_over_assets)		0.01	0.03***
		(0.01)	(0.02)
log(pat_stock_over_assets):log(linux_stock_over_assets)			0.01***
			(0.00)
Year dummy	*Yes*	*Yes*	*Yes*
R^2	0.59	0.60	0.60
Adj. R^2	0.50	0.50	0.50
Num. obs.	1044	1044	1044

$^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.1$

Table 5: Dependent variable $log(sales/assets)$. All equations include year dummy. Method of estimation is the fixed effect panel data analysis.

(Henkel and Baldwin, 2009).

The hybrid business model emphasis firms which are developing FOSS but at the same time also developing proprietary software. It is demonstrated that hybrid business model is an entry strategy but once the firm has the means to sustain within the market it does not change its business model to a pure FOSS or to a pure proprietary software development (Bonaccorsi et al., 2006; Dahlander and McKelvey, 2005; Bonaccorsi and Rossi, 2003). In our case, patenting and contributing to the Linux project show that both activities are carried out together. However, we cannot show whether there is a transition on the IPR model that the firms are adopting. This result show some similarities with the results for the faculty members who contribute to the scientific knowledge while filing patents at the same time (Fabrizio and Di Minin, 2008). Authors argue that both activities are complementary. In our research also we found that both opposite IPR approaches are in fact conducted at the same time which affects positively the sales performances.

The IP modularity concept have been developed for game software developers (Henkel and Baldwin, 2009). In this concept while software producers are reveling some of their codes, they also keep the crucial parts of their software as closed. The IP modularity helps firms to better capture value in situations where knowledge and value creation are distributed among many actors. It helps to reconcile co-creation and value capture. One of the typical examples are video card producers Nvidia and

ATI. These firms are contributing to the development of the Linux kernel but on the other hand they do not release their drivers' source code. Firms knowledge sharing is selective as it has been shown by Henkel (2006) for embedded Linux developers. It is very probable that firms in our sample also are expecting to have a positive image of the company within the community and the reduction of the cost of maintenance while revealing their code. Moreover, they are also creating an important user base for the utilization of their hardware.

5 Conclusion

This paper studies the interaction of the patenting and contribution to the Linux kernel project on sales value. It is found that patenting and contribution to an important FOSS project contributes positively to the sales value. Moreover, their interaction also contribute positively to the firm performance. The hybrid business model among FOSS developer firms and IP modularity have been treated within the literature and this paper discusses the results within this literature.

References

Allison, J. R., Lemley, M. A., and Walker, J. (2009). Extreme Value or Trolls on Top? The Characteristics of the Most-Litigated Patents. *University of Pennsylvania Law Review*, 158(1).

Amor, J., Robles, G., and Gonzalez-Barahona, J. (2005). Measuring woody: The size of debian 3.0. *arXiv preprint cs/0506067*, (004337):1–15.

Arora, A., Fosfuri, A., and Gambardella, A. (2001). Markets for Technology and their Implications for Corporate Strategy. *Industrial and Corporate Change*, 10(2):419–451.

Barthon, J. H. (2000). Reforming the Patent System. *Science*, 287(5460):1933–1934.

Bessen, J. (2009). Estimates of patent rents from firm market value. *Research Policy*, 38(10):1604–1616.

Bessen, J. (2010). The Two Faces of Technical Knowledge: Information Commodity and Human Capital.

Bessen, J., Ford, J., and Meurer, M. (2011). The Private and Social Costs of Patent Trolls. *Boston University School of Law Working Paper*, 11(45).

Bessen, J. and Hunt, R. M. (2007). An Empirical Look at Software Patents. *Journal of Economics & Management Strategy*, 16(1):157–189.

BlackDuckSoftware.com (2012). Open Source License Data.

Bonaccorsi, A., Giannangeli, S., and Rossi, C. (2006). Entry Strategies Under Competing Standards: Hybrid Business Models in the Open Source Software Industry. *Management Science*, 52(7):1085–1098.

Bonaccorsi, A. and Rossi, C. (2003). Why Open Source software can succeed. *Research Policy*, 32(7):1243–1258.

Corbet, J., Kroah-Hartman, G., and McPherson, A. (2012). Linux Kernel Development. Technical Report March, The Linux Foundation.

Dahlander, L. and Magnusson, M. G. (2005). Relationships between open source software companies and communities: Observations from Nordic firms. *Research Policy*, 34:481–493.

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

Dosi, G., Marengo, L., and Pasquali, C. (2006). How much should society fuel the greed of innovators? On the relations between appropriability, opportunities and rates of innovation. *Research Policy*, 35(8):1110–1121.

Fabrizio, K. R. and Di Minin, A. (2008). Commercializing the laboratory: Faculty patenting and the open science environment. *Research Policy*, 37(5):914–931.

Hall, B. H. (2004). Exploring the patent explosion. *ESRC Centre for Business Research, Unversity of Cambridge*, (291).

Hall, B. H. (2007). Patents. In *The New Palgrave: A Dictionary of Economics*, number September 2006. Palgrave Macmillan.

Hall, B. H., Griliches, Z., and Hausman, J. (1986). Patents and R&D: Is there a lag? *International Economic Review*, 27(2):265–283.

Hall, B. H. and MacGarvie, M. J. (2010). The private value of software patents. *Research Policy*, 39(7):994–1009.

Harison, E. and Koski, H. (2010). Applying open innovation in business strategies: Evidence from Finnish software firms. *Research Policy*, 39(3):351–359.

Henkel, J. (2006). Selective revealing in open innovation processes: The case of embedded Linux. *Research Policy*, 35(7):953–969.

Henkel, J. and Baldwin, C. Y. (2009). Modularity for Value Appropriation : Drawing the Boundaries of Intellectual Property.

Kortum, S. and Lerner, J. (1999). What is behind the recent surge in patenting? *Research Policy*, 28(1):1–22.

Levy, S. (2010). *Hackers: Heroes of the Computer Revolution*. O'Reilly Media, 25th anniv edition.

Mansfield, E., Schwartz, M., and Wagner, S. (1981). Imitation costs and patents: an empirical study. *The Economic Journal*, 91(364):907–918.

Mazzoleni, R. and Nelson, R. R. (1998). The benefits and costs of strong patent protection: a contribution to the current debate. *Research Policy*, 27(3):273–284.

Mowery, D. C. (2001). The growth of patenting and licensing by U.S. universities: an assessment of the effects of the Bayh Dole act of 1980. *Research Policy*, 30(1):99–119.

Nelson, R. R. (1994). Intellectual property protection for cumulative systems technology. *Columbia Law Review*, 94(8):2674.

Raymond, E. S. (1999). *The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly Media.

Rentocchini, F. (2011). Sources and characteristics of software patents in the European Union: some empirical considerations. *Information Economics and Policy*, 23(1):141–157.

Shapiro, C. (2001). Navigating the Patent Thicket: Cross Licenses, Patent Pools, and Standard Setting. In Jaffe, A. B., Lerner, J., and Stern, S., editors, *Innovation Policy and the Economy, Volume 1*, pages 119 – 150. MIT Press.

Stallman, R. M. (2002). *Free Software, Free Society: Selected Essays of Richard M. Stallman*. GNU Press.

Useche, D. (2012). Are patents signals for the IPO market? Evidence for an empirical analysis of the US and the European software industry. *DRUID*.

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