

**THE IMPLICATIONS OF INCUMBENT INTELLECTUAL  
PROPERTY STRATEGIES FOR OPEN SOURCE SOFTWARE  
SUCCESS AND COMMERCIALIZATION**

A Dissertation  
Presented to  
The Academic Faculty

by

Wen Wen

In Partial Fulfillment  
of the Requirements for the Degree  
Doctor of Philosophy in the  
School of College of Management

Georgia Institute of Technology  
August 2012

**THE IMPLICATIONS OF INCUMBENT INTELLECTUAL  
PROPERTY STRATEGIES FOR OPEN SOURCE SOFTWARE  
SUCCESS AND COMMERCIALIZATION**

Approved by:

Dr. Chris Forman, Advisor  
College of Management  
*Georgia Institute of Technology*

Dr. Marco Ceccagnoli  
College of Management  
*Georgia Institute of Technology*

Dr. Dan Breznitz  
College of Management  
*Georgia Institute of Technology*

Dr. Sandra Slaughter  
College of Management  
*Georgia Institute of Technology*

Dr. D.J. Wu  
College of Management  
*Georgia Institute of Technology*

Dr. Ram Chellappa  
Goizueta Business School  
*Emory University*

Date Approved: June 18, 2012

To Mom, Dad, and Xihao

## ACKNOWLEDGEMENTS

First of all, I sincerely thank Chris Forman for being a great advisor and mentor. This dissertation would not have been possible without his guidance. I owe him a great debt of gratitude for his generous support and boundless patience throughout the dissertation process. I am honored to follow in his footsteps to always strive to identify important research questions and conduct rigorous research. I am also indebted to the constructive advice and insights of my committee members, Sandra Slaughter, Marco Ceccagnoli, DJ Wu, Dan Breznitz, and Ram Chellappa. Their sage counsel along the course of this dissertation has both enriched my research in Information Systems and cultivated my interdisciplinary knowledge and skills.

I also gratefully acknowledge the enormous support from the entire faculty of the Information Technology Management group and the Scheller College of Business at Georgia Tech. I would like to specially thank Sri Narasimhan, Han Zhang, Saby Mitra, Eric Overby, Florin Niculescu, Lizhen Xu, Marie Thursby, Jerry Thursby, Stuart Graham, Ajay Kohli. Each one of them has provided me much guidance and precious knowledge at different stages of the Ph.D. program. I am very fortunate to have been in such an inspiring environment. I also thank the fellow students Peng Huang, Dezhi Yin, Chih-hung Peng, Tianshi Wu, Bin Li, German Retana, Hemang Subramanian, Michael Frutiger, Jiao Xu for being a group of outstanding colleagues.

My dissertation is also indebted to many invaluable suggestions from a variety of conferences and workshops. Special thanks are due to Ajay Agrawal, Tim Bresnahan, Alberto Galasso, Shane Greenstein, Josh Lerner, Megan MacGarvie, Timothy Simcoe,

Scott Stern, and participants at the 2012 NBER Patents, Standards, and Innovation Conference, the 2011 REER conference at Georgia Tech, the 2011 WISE, the 2011 CIST, the 2011 Kauffman Entrepreneurship Mentoring Workshop, the 2011 ICIS doctoral consortium, and the 2010 ICIS. My research also owes a great deal to the generous financial support from a Kauffman Dissertation Fellowship, a NBER Honorarium for the Study of Patents, Standards and Innovation, a Kauffman Foundation/Georgia Research Alliance grant for the study of entrepreneurship and productivity.

Finally, I wish to express my deep appreciation to my parents, Guiqiong Song and Xiaoning Wen, and my fiancé, Xihao He. Thank them for their unconditional love, sacrifice, and support during the entire doctoral program. No matter how challenging this journey is, every word and smile from them has always been the source of my happiness and courage. This dissertation is dedicated to them.

# TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENTS	iv
LIST OF TABLES	viii
LIST OF FIGURES	x
SUMMARY	xi
<u>CHAPTER</u>	
1 INTRODUCTION	1
2 THE IMPACT OF INTELLECTUAL PROPERTY RIGHTS ENFORCEMENT ON OPEN SOURCE SOFTWARE PROJECT SUCCESS	6
2.1 Introduction	6
2.2 Theoretical Motivation And Hypotheses	10
2.3 Econometric Approach	19
2.4 Measures	23
2.5 Empirical Models And Results	32
2.6 Conclusions	53
3 PATENT POOLS, THICKETS, AND OPEN SOURCE SOFTWARE ENTRY BY START-UP FIRMS	56
3.1 Introduction	56
3.2 Theory And Hypotheses	61
3.3 Research Setting	69
3.4 Data	72
3.5 Measures	74
3.6 Empirical Strategies And Results	81
3.7 Conclusions	103

APPENDIX A:	FOR CHAPTER 2: FURTHER DETAILS ON DATA CONSTRUCTION AND EXAMPLES OF PROJECTS IN THE TREATED GROUPS	105
APPENDIX B:	FOR CHAPTER 3: PROOFS IN SECTION 3.2	110
APPENDIX C:	FOR CHAPTER 3: IDENTIFICATION OF SOFTWARE SEGMENTS AND THE MATCHING PATENT CLASSES	113
APPENDIX D:	FOR CHAPTER 3: KEYWORDS USED TO IDENTIFY OSS ENTRY	120
REFERENCES		121

## LIST OF TABLES

	Page
Table 2.1: Summary of Cases	21
Table 2.2: Summary Statistics, Chapter 2	26
Table 2.3: User Interest, SCO v. IBM	37
Table 2.4: User Interest, FireStar/DataTern v. Red Hat	38
Table 2.5: User Interest (Alternative Sample), SCO v. IBM	40
Table 2.6: User Interest (Alternative Sample), FireStar/DataTern v. Red Hat	41
Table 2.7: User Interest, Sub-Sample	43
Table 2.8: Developer Activity, SCO v. IBM	47
Table 2.9: Developer Activity, FireStar/DataTern v. Red Hat	48
Table 2.10: Developer Activity (Alternative Sample), SCO v. IBM	50
Table 2.11: Developer Activity (Alternative Sample), FireStar/DataTern v. Red Hat	51
Table 2.12: Developer Activity, Sub-Sample	52
Table 3.1: Patent Pledging Events	71
Table 3.2: Summary Statistics, Chapter 3	80
Table 3.3: Pool Patents Compared to Non-pool Patents	82
Table 3.4: Conditional Fixed-effect Poisson	85
Table 3.5: GMM Estimates with IVs	90
Table 3.6: C-test, First Stage Results of OLS IV Regressions	93
Table 3.7: GMM Estimates Using Different Sets of IVs	94
Table 3.8: Falsification Test, Impact on Proprietary Software Products	95
Table 3.9: Conditional Fixed-effect Poisson, Add Two-way Interactions	99
Table 3.10: GMM Estimates with IVs, Add Two-way Interactions	101

Table 3.11: Conditional Fixed-effect Poisson, Add Two-way/Three-way Interactions	102
Table A.1: Examples in the treated group “Linux_kernel”	108
Table A.2: Examples in the treated group “object_relational”	108
Table A.3: Examples in the treated group “business_app_on_Linux”	109
Table A.4: Examples in the treated group “JBoss_related”	109
Table C.1: Examples of PROMT Codes	116
Table C.2: Examples of the PROMT-CorpTech Concordance	116
Table C.3: Examples of the Concordance between Segments and Patent Classes	117

## LIST OF FIGURES

	Page
Figure C.1: Identification of Software Segments	118
Figure C.2: Mapping Software Segments to Patent Subclasses	119

## SUMMARY

While open source software (OSS) emphasizes open access to the source code and avoids the use of formal appropriability mechanisms, there has been little understanding of how the existence and exercise of formal intellectual property rights (IPR) such as patents influence the direction of OSS innovation. This dissertation seeks to bridge this gap in prior literature by focusing on two closely related topics. First, it investigates how OSS adoption and production are influenced by IPR enforcement exercised by proprietary incumbents. Second, it examines the impact of royalty-free patent pools contributed by OSS-friendly incumbents on OSS innovation. It particularly looks at the influence of such patent pools on OSS product entry by start-up firms.

The first chapter provides an overview on the dissertation. The second chapter investigates how intellectual property rights (IPR) enforcement against open source software (OSS) affects the adoption and production of related OSS projects. This chapter suggests that when an IPR enforcement action is filed, user interest and developer activity will be negatively affected in two types of related OSS projects—those that display technology overlap with the litigated OSS and business projects that are specific to a focal litigated platform. These hypotheses are examined using two widely publicized lawsuits—*SCO v. IBM* and *FireStar/DataTern v. Red Hat*. The empirical analyses are based on data from SourceForge.net. The difference-in-difference estimates show that in the months following the filing of *SCO v. IBM*, OSS projects that exhibit high technology overlap with the litigated OSS experienced a 15% greater decline in user interest and 60% greater decline in developer activity than projects in the control group; OSS projects

that are intended for business and specific to a litigated OSS platform had a 34% greater decline in user interest and 86% larger decrease in developer activity than the control group. Similar results are also found following the filing of *FireStar/DataTern v. Red Hat*. These results are robust to a variety of robustness checks, including a falsification exercise and subsample analyses.

The third chapter examines whether open source software (OSS) patent pools contributed by large software incumbents influence new OSS product entry by start-up software firms. It argues that increases in the size of the OSS patent pool related to a software segment will facilitate OSS entry by start-up firms into the same segment; further, the marginal effect of the pool on OSS entry will be especially large in software segments where the cumulativeness of innovation is high or where patent ownership in a segment is concentrated. These hypotheses are empirically tested through examining the impacts of a major OSS patent pool—the Patent Commons, established by IBM and a few others in 2005—on OSS entry by 2,054 start-up firms from 1999 to 2009. The empirical results largely support these hypotheses and are robust to adding a variety of controls as well as to GMM instrumental variables estimation.

# CHAPTER 1

## INTRODUCTION

Open source software (OSS) has emerged as a new model of software innovation that draws upon public knowledge contributed by many individuals and commercial firms (von Hippel and von Krogh 2003). The success of OSS has been evidenced by the rapid growth of the Linux ecosystem that was worth \$25 billion as of 2008, the widespread adoption of Apache HTTP Server by 2.6 million newly established websites as of 2009, and the surge of the Android platform which hit a 60 percent market share for smartphone mobile operating system as of early 2012. Correspondingly, a body of literature has investigated the determinants of OSS success within the communities of developers and users (e.g. Stewart et al. 2006, Grewal et al. 2006) as well as how commercial firms appropriate value from open source innovation (e.g. West 2003, Fosfuri et al. 2008). It has also been emphasized that the driving force for the growth of OSS is its open access to the source code and the resulting freedom in modifying and advancing the software (Maurer and Scotchmer 2006).

However, one important issue that has been addressed little in prior literature is how the existence and exercise of formal intellectual property rights (IPR) such as patents play a role in influencing the direction of OSS innovation. The highly cumulative innovation process within the open source community, along with the difficulty in identifying source code ownership, makes OSS easily targeted by proprietary holders of IPR. This problem becomes even more acute when proprietary IPR holders face fierce competition from OSS and thus have incentives to dampen its diffusion, and when formal IPR have become increasingly valid and enforceable because of recent changes to US law. The best-known examples include the set of lawsuits against Linux by the SCO group and the recent cases against Android. One immediate question is how these IPR

enforcement actions affect the development of OSS. To address this question, it is useful to draw on theories on how the actions by proprietary holders of IPR can have implications for innovation under public domain (Heller and Eisenberg 1998, Gambardella and Hall 2006). However, most prior literature has focused on production of public knowledge goods such as scientific research (e.g., Walsh et al. 2003, Murray and Stern 2007, Walsh et al. 2007). There has been no systematic research examining the implications of formal IPR for the use and production of OSS, which might be different from other innovations under public domain in terms of the participation of commercial firms, greater market demand and commercial implications, and weaker reputation effects (von Hippel and von Krogh 2003, Gambardella and Hall 2006).

While no systematic research has quantified the effects of formal IPR on OSS, prior work (e.g., Cockburn and MacGarvie 2011) suggests that the existence of overlapping IPR, particularly patents (denoted as patent thickets), can create substantial costs for firms to commercialize software products in general. Several mechanisms promoting the sharing of collective IPR (i.e., patent pool, cross-licensing, standards-setting organizations) have been suggested to ameliorate this concern (Shapiro 2001). It remains to explore what collective rights mechanisms could be effective within the OSS communities and how. For example, since 2005, an increasing number of patent-pledging events have been announced by large OSS-friendly incumbents such as IBM, Sun Microsystems, and Nokia. Moreover, a few royalty-free patent pools such as the Patent Commons and Open Invention Network were established within the OSS communities, representing a special type of institution that offers royalty-free usage of patents to any OSS firm that promises not to sue the pool's beneficiaries (Lévêque and Ménière 2007, Serafino 2007, Hall and Helmers 2011). It is difficult to draw on the existing literature on traditional collective rights organization to evaluate the implications of those institutions. This is because prior studies have largely focused on institutions that delineate specific licensing rules and restrictions to outside members, whereas in this setting these

collective rights are royalty-free to any firm as long as it produces OSS. Thus, it is important to systematically investigate how such sharing and pledging of IPR affect OSS innovation.

My dissertation hopes to bridge these gaps in prior literature and takes the first step to study the implications of formal IPR for OSS. I particularly investigate the impacts of the following two types of IPR strategies exercised by incumbents: (i) IPR enforcement actions by incumbents, such as these lawsuits against a variety of OSS platforms as mentioned above, and (ii) the creation of royalty-free patent pools by OSS-friendly incumbents, which arguably serve as a counter measure against those litigation attacks. In short, my dissertation asks: how do these incumbent IPR strategies affect the direction of OSS innovation?

More specifically, in the second chapter, I investigate how IPR enforcement actions exercised by proprietary IPR holders affect OSS project success. I focus on two key metrics of OSS success: user interest and developer activity. The main hypotheses of this chapter are built upon prior research on OSS user interest (e.g., Subramaniam et al. 2009) and developer motivations (e.g., Lakhani and Wolf 2003, Roberts et al. 2006, Stewart et al. 2006). I argue that when an IPR enforcement action is filed, user interest and developer activity will be negatively affected in two types of OSS projects—those that have technology overlap with the litigated project and those that are intended for business and specific to a litigated OSS platform.

I test these hypotheses by examining how the filing of two widely publicized lawsuits—*SCO v. IBM* and *FireStar/DataTern v. Red Hat*—affected user interest and developer activity in these two types of projects. I use difference-in-difference estimation as the method of causal inference (Angrist and Pischke 2009) and extract data from the largest available repository of OSS projects—SourceForge.net (SourceForge) to empirically test the hypotheses. Following the existing literature (Subramaniam et al. 2009, Chengalur-Smith et al. 2010), I use the number of project downloads as a proxy for

user interest and the number of closed artifacts as a proxy of developer activity. I find that in the months following the filing of *SCO v. IBM*, OSS projects that exhibit high technology overlap with the litigated OSS experienced a 15% greater decline in user interest and 60% greater decline in developer activity than projects in the control group; OSS projects that are intended for business and specific to a litigated OSS platform had a 34% greater decline in user interest and 86% larger decrease in developer activity than the control group. I find similar results following the filing of *FireStar/DataTern v. Red Hat*. These results are also robust to a variety of robustness checks, including a falsification exercise and subsample analyses.

The negative effects of IPR enforcement on OSS use and production as identified in the second chapter provides some evidence that the “tragedy of the anti-commons” exists in the setting of OSS innovations. The third chapter investigates whether and how the creation and the size of a royalty-free OSS patent pool may positively encourage OSS innovation by mitigating the anti-commons problem. While OSS innovation can be examined from many different perspectives, my research strategy is to look at the impact of an OSS patent pool along a margin of innovation where it is likely to have the strongest impact. Therefore, the research focus of this chapter is on the software product entry behavior by start-ups using an OSS license (denoted as OSS entry).

Following prior work by Llobet (2003) and Galasso and Schankerman (2010), I first develop a model that shows that increases in the size of the pool influence the outcome of the litigation game and consequently the start-up’s OSS entry decision. Comparative statics from the model show that (i) changes in the size of the OSS patent pool related to a software segment facilitates OSS entry by start-up firms into the same segment; and (ii) the marginal effect of the pool on OSS entry will be especially large in software segments where the cumulativeness of innovation is high or where patent ownership in a segment is concentrated.

To empirically test these propositions, I focus on one major OSS patent pool—the Patent Commons—and assemble data on OSS entry using data on product releases from 2,054 start-up software firms contained in the Gale database “PROMT”. Then I follow prior work (Cockburn and MacGarvie 2009, 2011) and allocate patents to software product market segments. This mapping is used to identify the number of pool patents related to each market segment, as well as the cumulateness of innovation and patent ownership concentration in the segment. I employ count data conditional fixed effects models, so the empirical strategy is to look at whether time series variation in the size of the OSS patent pool related to a software segment is associated with changes in the number of OSS entrants into that segment.

The results from the third chapter suggest that a 10% increase in the size of an OSS patent pool related to a software segment is associated with a 1.5%-2.9% increase in the rate of OSS entry by start-ups into that segment. The marginal impact of the OSS patent pool is significantly greater in segments where the cumulateness of innovation is high: a 10% increase in the size of an OSS patent pool is associated with a 3.8%-5.6% increase in the rate of OSS entry when the cumulateness of innovation is at its 90th percentile, compared to no significant increase when cumulateness is at its 10th percentile. The effect of OSS patents pools is also greater when the concentration of patent ownership is high. A 10% increase in the size of an OSS patent pool is associated with a 1.3%-1.7% increase when patent concentration is at its 90th percentile and no increase when patent concentration is at its 10th percentile, however the statistical significance of these results vary somewhat across specifications. I also explore the robustness of these results by adding a variety of controls as well as GMM instrumental variables estimation.

**CHAPTER 2**

**THE IMPACT OF INTELLECTUAL PROPERTY RIGHTS**

**ENFORCEMENT ON OPEN SOURCE SOFTWARE PROJECT**

**SUCCESS**

**2.1 Introduction**

Open source software (OSS) has seen increasing adoption and development by both organizations and individuals in recent years, and in some markets provides the infrastructure for a significant share of overall economic activity. For example, based on a survey by Netcraft, Apache and its derivatives had been adopted by 2.6 million newly established websites in October 2009 (Netcraft 2009). An increasing body of literature (see Feller et al. 2005, Lerner and Tirole 2005a, Maurer and Scotchmer 2006, and von Hippel 2006 for recent reviews) has emerged in information systems and other fields studying, among many other things, the incentives for user contributions (e.g., Lakhani and von Hippel 2003, Hann et al. 2004, Roberts et al. 2006), firm strategies in open source (e.g., West 2003, Henkel 2006), and the impact of licensing and legal regimes (e.g., Lerner and Tirole 2005b, Stewart et al. 2006).

Despite the increasing popularity of OSS, the success of OSS projects vary widely—most cease to have activity after the first year (Stewart et al. 2006). In response to this, many researchers have studied the factors associated with OSS project success (e.g. Grewal et al. 2006, Stewart et al. 2006, Chengalur-Smith et al. 2010). For example, a recent stream of studies has focused on the role of project characteristics such as the social network of developers (e.g. Grewal et al. 2006), project sponsorship (e.g. Stewart et al. 2006), and network social capital (Singh et al. 2011) on project success.

The success of OSS projects is likely to be further influenced by the behavior of market participants who do not employ the OSS model. The development of new

knowledge is commonly organized around two types of models. The first, the so-called “private investment” or “proprietary” model, uses private investment to fund new projects and formal and informal intellectual property rights (IPR) protection to appropriate the returns from these investments (von Hippel and von Krogh 2003, Gambardella and Hall 2006). In the second type of model—the so-called “public domain,” “collective action,” or “private-collective action” models like open source—knowledge workers produce public goods that cannot be withheld from other users (e.g., von Hippel and von Krogh 2003). It has been widely theorized that actions by individuals and organizations using the first model can have implications for those using the second (Heller and Eisenberg 1998, Gambardella and Hall 2006). In particular, enforcement of formal IPR like patents may discourage production and use of public knowledge goods like OSS and may be another example of the “tragedy of the anti-commons” that has arisen in other collective action models such as scientific research (Gambardella and Hall 2006).

While these ideas hold great salience for the success of OSS projects, they have to my knowledge not been systematically empirically tested. Recent research has explored the implications of the introduction of formal IPR on the production and use of knowledge in scientific research, demonstrating mixed results (e.g., Walsh et al. 2003, Murray and Stern 2007, Walsh et al. 2007). However, results for OSS may be different from these other contexts because of differences in institutional characteristics such as greater market demand, the presence of non-producing consumers, and weaker reputation effects (Gambardella and Hall 2006).

With these ideas in mind, I take a first step towards empirically assessing the implications of formal IPR for the production and use of OSS. I motivate my hypotheses using prior theoretical work that has demonstrated how the possession and exercise of formal IPR can lead to an under-provision of knowledge under the public domain model. Using prior research on OSS user interest (e.g., Subramaniam et al. 2009) and developer

motivations (e.g., Lakhani and Wolf 2003, Roberts et al. 2006, Stewart et al. 2006), as a starting point, I theorize the implications of formal IPR for two key metrics of OSS success: *user interest* and *developer activity*. I focus on the implications of IPR *enforcement* through litigation actions rather than the existence of formal IPR like patents or copyrights because the former provides a more proximate signal about the risks to users and developers. I next develop the main hypothesis of this study: I argue that when an IPR enforcement action is filed, user interest and developer activity will be particularly affected in two types of OSS projects—those that have technology overlap with the litigated project and those that are intended for business and specific to a litigated OSS platform.

I test these hypotheses by examining how the filing of two widely publicized lawsuits—*SCO v. IBM* and *FireStar/DataTern v. Red Hat*—affected user interest and developer activity in these two types of projects. My method of causal inference is difference-in-difference estimation (Angrist and Pischke 2009). I use data from the largest available repository of OSS projects—SourceForge.net (SourceForge). I first examine the implications for user interest by studying the impact of IPR enforcement actions on the number of project downloads. I find that in the months following the filing of *SCO v. IBM*, monthly downloads of OSS projects that display technology overlap with the focal litigated OSS fell 15%-16% relative to a control group, while those projects that are intended for business and specific to a litigated OSS platform fell 34%-35% more than other projects. I find similar results following the filing of *FireStar/DataTern v. Red Hat*. These results are robust to a large variety of analyses and additional tests.

I next examine the effects of the same IPR enforcement actions for developer activity. In keeping with prior literature (e.g., Chengalur-Smith et al. 2010), I use the number of closed artifacts in a project as a proxy. In the months following the filing of *SCO v. IBM*, the number of closed artifacts per month for projects that exhibit technology overlap with the litigated one fell 60.2%-63.8% more than the control group; business-

related projects specific to the litigated platform also had 86%-87% fewer closed artifacts per month following the suit than the control group. I find similar results when I study the number of closed artifacts following the filing of *FireStar/DataTern v. Red Hat*. Once again, to address concerns about time-varying omitted variables, I further examine the robustness of these results using a series of additional tests.

As noted above, my research advances recent literature on the implications of formal IPR for the generation and use of new public good knowledge. However, my findings also have implications for other research areas. In particular, my research advances recent literature that has studied how outside factors such as the social network in which an OSS project is embedded (e.g., Grewal et al. 2006, Singh 2010) and the niche<sup>1</sup> size in which a project operates (e.g. Chengalur-Smith et al. 2010) influence an OSS project's likelihood of success. In particular, as suggested by Chengalur-Smith et al (2010), an individual project's success is not only determined by the characteristics of the project but also by the characteristics of the project niche and its environmental process. I follow this line of research and stress the enforcement of IPR as one important environmental factor influencing OSS success.

Last, my study has important implications for the study of IPR in software. The validity and enforceability of software patents has become more certain in US law since 1980, and individual and firm behavior has changed accordingly (Graham and Mowery 2003). The increasing use of software patents has contributed to a concern about their detrimental effect on the development (Scotchmer 1991, Bessen and Maskin 2009) and adoption (Tucker 2012) of new software. Empirical work in this area has usually focused on the implications of IPR for the behavior of firms producing software under the proprietary model; in particular, one line of work has focused on the effects of formal

---

<sup>1</sup> A niche is defined from the organizational ecology perspective as a set of organizations that have something in common (Chengalur-Smith et al. 2010).

IPR on new product entry by software firms (e.g., Cockburn and MacGarvie 2011). My study adds to this stream of literature by providing empirical evidence about the implications of IPR protection for software development activity under the OSS model (Samuelson et al. 1994, Cohen and Lemley 2001, Graham and Mowery 2005).

## **2.2 Theoretical Motivation And Hypotheses**

The design of systems to provide incentives for researchers and other knowledge producers has long been an active area of research (e.g., Scotchmer 1991). Scholars have described multiple regimes to allocate resources for the generation of new knowledge (e.g., von Hippel and von Krogh 2003). One system relies on private investment by individuals and organizations to produce private goods and knowledge. To provide appropriate incentives for private firms and individuals to invest, new knowledge produced under this system is often protected by traditional regimes of intellectual property protection like patents and copyrights. This regime has commonly been labeled the “private investment” or “proprietary research” model (e.g., von Hippel and von Krogh 2003, Gambardella and Hall 2006).

Under the second system, knowledge outputs are public goods; if any user consumes them, they cannot be withheld from other users (e.g., Olson 1971). In contrast to the private investment model, rewards under this model are provided indirectly through, for example, increased reputation, access to additional research resources, and potential subsequent financial returns in the form of salary, prizes, and so forth (e.g., Merton 1957, von Hippel and von Krogh 2003, Gambardella and Hall 2006). This model has alternatively been labeled the “open science”, “collective action”, or “public domain” model by researchers (Merton 1957, von Hippel and von Krogh 2003, Gambardella and Hall 2006). In its emphasis on the production of public goods and avoidance of the use of formal and informal appropriability mechanisms, this latter model shares many characteristics with the “private-collective action” model that has been used to describe

OSS development (von Hippel and von Krogh 2003). In keeping with prior work on OSS that motivates my research (Gambardell and Hall 2006) and my focus on differences in the use of appropriability mechanisms (such as patents and copyrights) across regimes, for the purposes of this discussion I group these two models together and label them the “public domain” model.

A long literature has explored how the introduction and enforcement of IPR under the proprietary model may promote the development and diffusion of knowledge under the public domain model. For example, patents may promote complementary investments by firms to commercialize new knowledge. The most common example of this is when scientific knowledge needs downstream investments to become commercial products (e.g., Thursby et al. 2001). Patents may also facilitate the effective functioning of markets for technology (e.g., Arora et al. 2001). In contrast, other research has emphasized how the introduction and enforcement of IPR increases the costs of using public knowledge that previously had been used freely. This view has been described as an “anti-commons” effect: since the imposition of formal IPR can exclude others from the use of new ideas, it can inhibit the cumulative nature of new knowledge production both by increasing the costs of using protected knowledge and the costs of building upon it (e.g., Heller and Eisenberg 1998).

Prior work that has evaluated the anti-commons effect in other settings has explored the implications of IPR both for the use of (previously public good) knowledge as well as the production of new knowledge (e.g., Walsh et al. 2003, Murray and Stern 2007, Walsh et al. 2007). One theme from this literature is that the existence and exercise of IPR can increase the costs of using new knowledge inputs through the enforcement of licensing fees, terms of exclusivity, and infringement liability (e.g., Dasgupta and David 1994, Walsh et al. 2007). The research of Gambardella and Hall (2006) suggests that the anti-commons effect may be particularly important in the context of OSS development, however they neither directly model nor test this hypothesis. I advance this prior work by

developing specific hypotheses about how IPR enforcement can influence user interest and developer activity in an OSS project.

### **2.2.1 How Does IPR Enforcement Affect User Interest?**

In this section, I investigate how IPR enforcement could affect the success of OSS projects in attracting user adoption (I label this as *user interest*). I focus on how an increase in the perceived costs to adoption through litigation risks translates into lower user interest (Stewart et al. 2006). I motivate this hypothesis through the technology adoption literature, particularly the Rank Approach described by Stoneman (2002). The rank approach asserts that an individual or organization will adopt a new technology (in my case, an OSS project) when the gross benefits to adoption exceed the costs.

Given the heterogeneity in OSS project characteristics, I expect differences across projects in the effects of IPR enforcement on user interest. In the remainder of this section, I describe the two types of projects for which user interest is likely to be particularly affected.

#### How Does IPR Enforcement Affect User Interest In OSS Projects That Exhibit Technology Overlap With The Litigated Technology?

The litigation externalities literature (e.g., Viscusi and Hersch 1990, Govindaraj et al. 2007) suggests that given an enforcement action targeting a single firm, other firms in the same industry may incur significant capital losses if the enforcement matures into litigation. The reason is that investors and customers may “view lawsuits against any firm in an industry as likely to increase the likelihood that all firms in the industry will be implicated in future lawsuits” (Prince and Rubin 2002). In my setting, an IPR enforcement action against the focal OSS could influence a user’s perception of whether similar OSS projects will become targets in the future. If this happens, a potential user would anticipate an increase in associated costs (e.g., anticipated royalties demanded by

IPR holders<sup>2</sup>, direct litigation costs<sup>3</sup>) that would reduce her overall utility from adopting software from the project. In fact, adopters have been defendants in several prominent OSS cases (e.g., *SCO v. DaimlerChrysler* and *SCO v. AutoZone*).

When a potential user assesses whether her intended OSS project will be affected, she will use cues such as the software's observable attributes to assess the likelihood of infringement risk. This is because software is a complex technology consisting of many independently protectable inventions. The existence of many anonymous developers in the OSS community makes it difficult to identify the provenance of the code and therefore assess the infringement risks (Lerner and Tirole 2005a).

Here I focus on one software attribute that serves as a cue for potential users to determine whether expected costs will increase as a result of a litigation event: technology overlap with the focal litigated technology. Following existing literature, technology overlap is considered to be high when two technologies draw from related technological areas (e.g., Lanjouw and Schankerman 2001). Technology overlap will generally be higher when new inventions combine existing conceptual materials such as routines or technologies (Nelson and Winter 1982) or constituents of invention along the lines of what Schumpeter calls "factors" (Schumpeter 1939, Fleming 2001). In computer

---

<sup>2</sup> For example, following the initiation of the *SCO v. IBM* case, SCO sent notices to over 1,500 corporate Linux users, requesting a roughly \$700-per-server license fee and stating that those refusing this offer could face litigation (Goetsch 2004). Royalty payments often arise from negotiations between a patentee and alleged infringer. For example, Red Hat set aside \$43 million for the settlement of IPR litigation claims related to JBoss (Mueller 2011)

<sup>3</sup> IPR enforcement actions seldom mature into litigation, and more likely settle with a royalty payment by the accused infringer to the IPR owner (Farrell and Merges 2004). The high cost of IPR litigation—estimated for patent suits to average between \$0.7-5.5 million per side—are a major determinant to settlement, even when the royalty demanded is out of proportion to the incremental value of the technological contribution. But the enforcement stance of the IPR owner can nevertheless drive the accused infringer to suffer these litigation costs. Courts will often use injunctions to prevent accused infringers from conducting any business that uses the patented technology (Lanjouw and Lerner 2001), thus forcing production lines to close or products to be removed from store shelves (Graham and Mowery 2003). This business risk, when coupled with high litigation costs, can drive accused infringers to settle on unfavorable royalty terms, thus driving up costs associated with IPR enforcement actions for adopters.

software, programmers can combine existing factors such as algorithms, methods, or specific lines of code. While technology overlap can be assessed for proprietary technologies through the use of bibliometric information in patents (e.g., Fleming 2001), in my setting potential users can assess technology overlap between two OSS projects by investigating project descriptions, key words, and documentation.

I argue that potential users will perceive higher potential litigation costs for projects with greater technology overlap. As a result, they may choose not to adopt. There are at least three reasons for why the perceived litigation risks may be higher for this group. First, OSS developers reuse knowledge and code extensively in order to efficiently integrate functionalities and reduce development costs (Haefliger et al. 2008). When the technology overlap between two OSS projects is high, it is even more likely that developers will reuse knowledge from the other project since technology reuse is most effective when developers are searching in a set of familiar components (Fleming 2001). Therefore, such projects will be more likely to have infringing code and for the use of infringing code to be identified. Further, because IPR enforcement strategies are commonly run in series, with early cases being used to build findings and evidence for later enforcement actions against others, the logic of transaction costs suggests that software with characteristics similar to the focal target would be more prone to subsequent enforcement. Moreover, IPR owners may have incentives to signal to a group of similar projects the intention to enforce their property rights so as to build a reputation for aggressive IPR protection (Lanjouw and Schankerman 2001) and I suggest that such a reputation would be most valuable in a technologically proximate space to the IPR holders' assets. Therefore, I argue an IPR rights holder is more likely to first target OSS projects in fields that display technology overlap.

Simply put, in the wake of an IPR enforcement action against a particular piece of OSS, potential adopters of OSS projects that overlap with the technology of the focal

litigated OSS may perceive an increase in the expected costs of use. Thus, I expect all else equal that enforcement of IPR will decrease a user's interest in these projects<sup>4</sup>.

*H1: IPR enforcement has a negative impact on user interest in OSS projects that have technology overlap with the focal litigated OSS.*

## How Does IPR Enforcement Affect User Interest In OSS Projects Specific To A Litigated Platform?

Investments by buyers and sellers in computer software often take place within the context of platforms: a set of interchangeable IT components that enable buyers and sellers to coordinate on the same technological advancement (Bresnahan and Greenstein 1999). As has been highlighted elsewhere, the OSS development methodology can have significant advantages for the creation and dissemination of platforms. The provision of complementary applications specific to a platform<sup>5</sup> is often controlled by formal, published interfaces that are proprietary. In contrast, in OSS third-party developers can add their own interfaces unanticipated by the original authors of the platform (West 2003). This can increase incentives for third-party development. Thus, many large and important OSS projects serve as platforms. This includes operating system projects like Linux as well as many other projects like Apache or Red Hat's JBoss enterprise middleware.

A large body of economic theory has emerged to explain buyer and seller behavior within the context of platforms. A central idea in much of this work is to explain the persistence of standards, which stand at the heart of computer platforms. Bresnahan

---

<sup>4</sup> I note the possibility that some users may switch from the focal (litigated) project into alternative goods with similar features. In that sense, my results can be viewed as an upper bound (i.e., conservative) estimate of how the costs of litigation influence user behavior.

<sup>5</sup> For simplicity, sometimes I denote those complementary applications that are specific to a platform as platform-specific software.

and Greenstein (1999) detail how the theory of standards makes two assumptions about buyer and seller behavior. First, buyers and sellers of IT make platform-specific investments. Second, some of these platform-specific investments are long-lived: in particular, they influence the future market choices available to buyers. These assumptions imply that buyers of computer software frequently make long-lived platform-specific investments that tend to sustain platforms for long periods. This will particularly be the case for computer software investments made by businesses, where the need to coordinate investment activity over time is greatest (Bresnahan and Greenstein 1996, 1999).

Prior work has demonstrated that when the costs of these long-lived platform-specific investments are high, buyers will delay adoption of both the platform and complementary software specific to the platform (Forman 2005). Further, the extent of delay is likely to be magnified when there exists uncertainty surrounding the potential success of the platform. In the presence of such uncertainty, buyers may delay adoption of the platform and complementary software until the uncertainty is resolved. If IPR enforcement targets a platform, it could therefore increase uncertainty surrounding the success of the platform and therefore delay adoption of both the platform and all platform-specific software.

This is particularly true because of the importance of network effects to platforms. Successful platforms require a critical mass of adopters and complementary software specific to the platforms. In the presence of positive feedback, platforms must get ‘over the hump’ to become a new standard (e.g., Bresnahan and Greenstein 1999, Farrell and Klemperer 2007). Legal actions that increase uncertainty about the future viability of the platform will make it that much more difficult for the platform to get ‘over the hump’, experience increasing returns to adoption, and survive. This will make potential buyers more reticent to make long-lived platform-specific investments.

Thus, IPR enforcement actions that target an OSS platform will likely have a negative impact on software projects that are specific to the platform, typically those for which software investments require long-lived investments as is the case for business software.

*H2: IPR enforcement has a negative impact on the user interest in business-related OSS projects that are specific to a focal litigated OSS platform.*

### **2.2.2 How Does IPR Enforcement Affect Developer Activity?**

In this section, I investigate for what types of projects IPR enforcement is likely to influence developer activity. A body of OSS literature highlights how intrinsic motivation and extrinsic motivations influence developers to participate in OSS projects (e.g. Lakhani and Wolf 2003, Roberts et al. 2006). Developers driven by intrinsic motivations participate for the pure enjoyment of coding (Roberts et al. 2006), sharing, and the free and open software ideology (Shah 2006). On the contrary, developers driven by extrinsic motivation participate more for its instrumental value (Ryan and Deci 2000): developers choose to participate only if the expected benefits are greater than the costs (e.g. the opportunity costs of their time) (Lerner and Tirole 2002). This latter set of motivations can further be classified into the benefits of direct monetary compensation; utility-based need for particular software; ego enhancement or peer recognition; career concern or skill improvement; potential commercialization interest or access to venture capital (Maurer and Scotchmer 2006).

I build upon prior literature that studies developer motivations by studying how differences in project characteristics influence extrinsic motivations through the costs and benefits of contributing (e.g., Lerner and Tirole 2002, 2005a, Gambardella and Hall 2006, Stewart et al. 2006). Namely, in parallel to the development of Hypotheses 1 and 2, I study how IPR enforcement actions affect the costs and benefits of developer motivations for projects that (1) show technology overlap with a litigated project or that (2) are

business applications specific to a litigated OSS platform. I focus on three implications of IPR enforcement for these two groups: (1) the costs of participating in the project; (2) the benefits of contributing to the project for self-use; and (3) the career-related benefits of contributing to OSS projects.

In the wake of IPR enforcement, developers' costs of participating in projects with technology overlap with the litigated one will increase. As noted above, such projects will commonly share code with the litigated project. IPR enforcement actions may therefore force developers working on those projects to spend additional effort to identify and search for the provenance of code and attest that they have the right to use a particular piece of code in the project (Lerner and Tirole 2005a).

IPR enforcement actions will also affect the benefits of contributing to these two types of projects. For one, a common reason that developers contribute to a project is to use the product they develop (e.g., von Hippel and von Krogh 2003, Stewart et al. 2006). Thus, developer activity and user interest are frequently correlated (Krishnamurthy 2002). As noted above, IPR enforcement actions will reduce user interests in these two types of projects, and so will similarly mute development incentives.

Further, developers who work on OSS projects for career-related signaling concerns will prefer to work on projects with a greater user base (Lerner and Tirole 2002). Greater user interest will translate into a wider audience for contributors that will make their efforts more visible (Parker and van Alstyne 2005). Thus, there are greater reputation benefits from working on popular projects (Lerner and Tirole 2002). If user interest declines as hypothesized in H1 and H2, this will lead to a similar decline in developer incentives. Similarly, many developers are employed by firms attempting to commercialize OSS projects (e.g., Lerner and Tirole 2002, 2005a). If user interest in these two types of projects decline, then commercialization and other indirect profit opportunities from these projects will decline as well. Weakening firm sponsorship in the

project and support for developers may lead to a decline in developer activity. These discussions lead to the following hypotheses:

*H3: IPR enforcement has a negative impact on developer activity in OSS projects that have technology overlap with the focal litigated OSS.*

*H4: IPR enforcement has a negative impact on developer activity in business-related OSS projects that are specific to a focal litigated OSS platform.*

### **2.3 Econometric Approach**

The method of causal inference I use is difference-in-difference estimation (Angrist and Pischke 2009). The basic idea of difference-in-difference estimation is to examine a set of treated units before and after the treatment (in this case, an IPR enforcement action). In my setting, the treated units are those OSS projects that show technology overlap with the litigated OSS or belong to business-related OSS projects specific to a litigated OSS platform. Difference-in-difference estimation has been used frequently in the IS literature as a method of causal inference, particularly in circumstances where I wish to examine how an exogenous event such as new firm entry (Forman et al. 2009), new movie broadcast (Smith and Telang 2009), or legal action (Tucker 2012) influences economic behavior such as purchase decisions. Difference-in-difference has also frequently been used in the literature evaluating the implications of IPR for production and use of knowledge under the collective action model (e.g., Murray and Stern 2007, 2008, Thompson et al. 2011). As in those papers and more broadly within the difference-in-difference approach, I used revealed preferences to test hypotheses. That is, I do not directly observe how the costs and benefits of OSS use and OSS development activity change after the litigation event. Instead, I use changes in use and development activity to infer that the net benefits from participating in these activities have changed. My use of difference-in-difference estimation requires me to estimate separate regression models for user interest and developer activity.

Indexing units by  $i$  and time by  $t$ , I adopt the basic form:

$$\begin{aligned} Outcome_{it} = & \alpha + \delta TreatmentGroup_i + \beta_0 AfterTreatment_t + \beta TreatmentGroup_i * \\ & AfterTreatment_t + \gamma RegressionControls_{it} + \varepsilon_{it} \end{aligned} \quad (2.1)$$

By plugging in zeros and ones for the binary variables in equation (2.1), the difference across groups in the before and after treatment is  $\beta$ . If  $\beta$  is negative, the treatment can be interpreted as having a negative effect on the outcome. Just as in a true experiment, this approach means that I can see whether behavior in the treatment group changes differently from that in the control group. In my setting, I examine the before and after performance of two treatment groups—OSS projects that show technology overlap with the litigated projects and business OSS projects that are specific to a litigated OSS platform.

I note that the parameter  $\beta_0$  may also be of interest. That is,  $\beta_0$  captures changes in user interest or developer activity across all projects after IPR enforcement. However,  $\beta_0$  may also reflect general time trends in user interest or developer activity (for example, because of some seasonality effect) and so in the absence of a control group it is difficult to identify the effects of the treatment from alternative time-varying factors. While I report estimates for  $\beta_0$ , I do not focus on them. In the next several sections I describe the treatment, how I identify the treatment groups, my regression controls, and several additional details about my experiment.

### **2.3.1 Cases of IPR Enforcement: The Treatment**

I choose well-publicized lawsuits as cases of IPR enforcement. Based on a search of major news outlets, I found six lawsuits as summarized in Table 2.1.

**Table 2.1 Summary of Cases**

Case number	Plaintiff	Defendant	Filing Date	Settlement Date	Litigated IPR	Technologies in dispute
1	SCO	IBM	3/7/2003	N/A	Copyright	Linux operating system
2	FireStar and DataTern	Red Hat	6/26/2006	6/11/2008	Patent	A method of interfacing an object-oriented software application with a relational database
3	NetApp	Sun	9/5/2007	N/A	Patent	7 patents related to NetApp's WAFL and RAID technology
4	IP innovation	Red Hat and Novell	10/15/2007	N/A	Patent	3 US patents covering the user interface in multiple workspaces sharing display system objects
5	Microsoft	TomTom	2/25/2009	3/30/2009	Patent	Five patents related to car navigation technology and three for file management systems
6	Software Tree	Red Hat, HP, Genuitec, and Dell	3/3/2009	N/A	Patent	A System and Method for Exchanging Data and Commands Between an Object Oriented System and Relational System.

Note: 1) Based on a search of major news outlets, I found six large, well-publicized lawsuits as cases of IPR enforcement against open source software and summarize them in the following table. The main reason for me not to focus on case 3 and case 4 lies in that case 4 was filed only one month after case 3 and thus it is difficult to disentangle potential adopters' behavioral changes in response to case 3 from their reactions to case 4. I do not choose case 5 and case 6 also for the same reason. Meanwhile, for case 5, the time window between filing date and settlement date is very short, so it is difficult to identify any behavioral changes. 2) N/A means that the case has not yet been settled.

While many papers implementing difference-in-difference estimation examine the implications of one treatment only, I choose to examine the implications of two treatments to improve the internal validity of my study (i.e., by showing my results hold in multiple contexts, this provides further confidence that my results do not reflect unobserved factors influencing the outcome of the treatment group) and its external validity (to suggest that my results reflect more than the outcomes around a particular IPR enforcement case). In particular, I focus on the initial announcements of two lawsuits: *SCO v. IBM* and *FireStar/DataTern v. Red Hat*.

My choice of lawsuits is guided by several factors. First, I hope to choose lawsuits that were contemporaneously viewed as economically important. These two cases have had wide news coverage<sup>6</sup> and also the defendants in both cases are commercial players that have significant capital investments in OSS. Second, my approach requires that the open source community not be aware of the risks of litigation ex ante (otherwise they will have made the behavioral adjustments that I posit in advance of the IPR enforcement announcement). As a result of this requirement, I do not examine any of the numerous follow-on announcements and lawsuits that followed the *SCO v. IBM* case. Last, I exclude cases for which the time between the filing of the IPR enforcement action and the settlement of the case was too short to observe any significant behavioral changes.

#### Case 1: *SCO v. IBM* (March 7, 2003)

On March 7, 2003, SCO filed a \$1 billion lawsuit against IBM. SCO asserted it had ownership of Unix and all of its derivative works, and some of its copyrighted Unix software was wrongly copied into Linux by IBM (SCO 2003). The lawsuit attracted

---

<sup>6</sup> By wide news coverage, I mean that many news outlets reported on the lawsuits.

significant publicity, partly because it was the first major IPR enforcement suit targeting OSS usage and development and partly because it related to Linux platform and so had important implications for the entire Linux ecosystem. Based on a Lexis-Nexis search I found that many major news outlets such as the Wall Street Journal, the San Jose Mercury News (San Jose, California), the Boston Globe, the Los Angeles Times, the Daily Telegraph (Sydney, Australia), and the Business Times (Singapore) all reported on this lawsuit around the filing date.

#### Case 2: *FireStar/DataTern v. Red Hat* (June 26, 2006)

The second major case, *FireStar/DataTern v. Red Hat*, is related to the JBoss Platform, the most popular Java Enterprise Edition-based platform that has attracted more than 600 certified ISV applications as of 2012 (Stevens 2012). On June 26, 2006, FireStar and DataTern filed a lawsuit asserting that Red Hat's JBoss suite – particularly Hibernate 3.0 – infringes FireStar's U.S. Patent No. 6,101,502. This patent details a method of interfacing an object-oriented software application with a relational database (Heubner et al. 2000). As many OSS observers noted, this lawsuit “is expected to take center stage on the legal front as SCO's copyright claims against IBM fade” and “is potentially more significant than the SCO case because it's about a patent that covers a basic concept or idea, not an expression of an idea, which copyright covers” (Rooney 2006). In the week following the lawsuit, Red Hat's stock fell substantially and closed at \$23.20 on July 3 2006, a drop of 28 percent from its six-year high of \$32.27 on May 8, 2006 (Krishnan 2006).

## **2.4 Measures**

### **2.4.1 Sample**

My primary data source is the SourceForge Research Data Archive (SRDA) (Van Antwerp and Madey 2008), which receives monthly database snapshots from

SourceForge. SourceForge is the largest repository of OSS—over 230,000 projects and over 3 million users and developers were registered at the time of my data collection (SourceForge 2009). SRDA provides more complete data on a monthly than on a weekly basis so I use project-month as my unit of analysis.

The sampling period for *SCO v. IBM* is from January 2002 to July 2003, with 14 months before *SCO v. IBM* and 5 months after. My choice of time window was influenced by several considerations. First, the extended period helps me to control for yearly and monthly time effects, in particular seasonality that may influence user and developer activity. Second, I choose 5 months after the lawsuit because another case was filed in August 2003, *Red Hat v. SCO*, that may shift how users and developers perceive the costs and benefits of interacting with OSS projects.<sup>7</sup> The sampling period for *FireStar/DataTern v. Red Hat* is from July 2005 to November 2006, with 12 months before *FireStar/DataTern v. Red Hat* and 5 months after. Because of constraints associated with my data source, I am not able to get data before July 2005. Also, I believe the window size of 5 months for *SCO v. IBM* and *FireStar/DataTern v. Red Hat* is long enough for me to capture the post-IPR enforcement reaction from the OSS community. I have experimented with changing the sampling periods and my results were robust to these changes.

I first construct my samples to test the implications of IPR enforcement for user interest. As noted by Rainer and Gale (2005) and Hahn et al. (2008), many registered OSS projects on SourceForge are “impulse” projects. For example, such projects may be final student projects or the results of developer experimentation. The quality of these projects is relatively low, so such projects are unlikely to reflect the tradeoffs in which I

---

<sup>7</sup> On August 4 2003, Red Hat, a major Linux distributor, filed a lawsuit against SCO, asserting that it was making “unfair, untrue and deceptive claims that Red Hat’s version of the open-source system contains code stolen from SCO” (Takahashi 2003)

am interested. To mitigate this problem and focus on active projects, I construct a baseline sample composed of projects that have positive downloads for each month during the sample.<sup>8</sup> The resulting panels have 3,928 OSS projects over 19 months for *SCO v. IBM* and 24,301 projects over 17 months for *FireStar/DataTern v. Red Hat*.

To examine how IPR enforcement affects developer activity, I first restrict my sample to projects that have total number of closed artifacts larger than zero over the sample period. I then further restrict this sample by only including projects above the 25th percentile of the distribution of total number of closed artifacts. This selection criterion is in line with other papers on developer activity suggesting that projects with very few artifacts may not have an active developer community and that the project's purpose is primarily to make existing source code public (rather than providing a forum for new contributions) (Chengalur-Smith et al. 2010)<sup>9</sup>. This rule results in a panel of 904 OSS projects over 19 months for *SCO v. IBM* and a panel of 2,311 projects over 17 months for *FireStar/DataTern v. Red Hat*.

## 2.4.2 Variables

In the following subsections I describe my dependent and independent variables. Summary statistics for all variables in the baseline sample are shown in the below table 2.2.

---

<sup>8</sup> I also constructed an alternative sample using projects that have positive downloads for at least one month during the sample period. The results are qualitatively similar to those using the baseline sample.

<sup>9</sup> I constructed an alternative sample using projects that have total number of closed artifacts over the sample period larger than zero. As shown in the Appendix, the results from this alternative sample are very consistent with those using the baseline sample.

**Table 2.2 Summary Statistics, Chapter 2**

		Variable	Mean	Std Dev.	Min	Max	Obs
<i>SCO v. IBM— User Interest</i>	Project- month-level	downloads	1927.211	16776.25 0	1	1120505	74632
		developers	3.754	5.613	1	132	74632
		cvs_commits	25.811	118.478	0	5169	74632
	Project- level	Linux_kernel	.017	.128	0	1	3928
		business_app_on_Linux	.004	.066	0	1	3928
		installed_base	6.762	2.786	0	15.140	3928
		restrictive_license	.707	.455	0	1	3928
<i>SCO v. IBM —Developer Activity</i>	Project- month-level	closed_artifacts	5.437	20.884	0	962	16272
		msg_posted	14.154	103.454	0	4375	16272
	Project- level	Linux_kernel	.023	.151	0	1	904
		business_app_on_Linux	.002	.047	0	1	904
		installed_base	8.224	2.020	3.4	15.14	904
		restrictive_license	.645	.479	0	1	904
<i>FireStar/Data Tern v. Red Hat—User Interest</i>	Project- month-level	downloads	1672.329	45668	1	8447172	413117
		developers	2.992	5.784	0	350	413117
		new_files	.304	3.720	0	775	413117
	Project- level	object_relational	.002	.042	0	1	24301
		JBoss_related	.001	.037	0	1	24301
		installed_base	7.099	2.108	.690	18.420	24301
		restrictive_license	.111	.314	0	1	24301
<i>FireStar/Data Tern v. Red Hat — Developer Activity</i>	Project- month-level	closed_artifacts	3.600	15.584	0	839	36976
		msg_posted	409.916	2259.681	0	68131	36976
	Project- level	object_relational	.002	.041	0	1	2311
		JBoss_related	.002	.046	0	1	2311
		installed_base	8.640	2.473	2.7	17.58	2311
		restrictive_license	.673	.469	0	1	2311

Notes: Number of observations differs across IPR enforcement actions. Variables with the label “Project-month-level” vary by project month, while those with the label “Project-level” vary only by project. Number of observations for “Project-level” variables is equal to the number of projects.

### 2.4.2.1 Dependent Variables

*User Interest.* I follow prior literature in using downloads as a market-based measure of popularity and user interest (e.g., Grewal et al. 2006, Subramaniam et al. 2009). As Grewal et al. (2006) note, when software projects are freely available, researchers have in the past used downloads as proxy for sales (e.g., Chandrashekar et al. 1999). Monthly downloads may deviate from adoption, however. Some potential users may download OSS without using it. Further, some fraction of downloads are from OSS hobbyists who are interested simply in looking at the source code and do not plan to adopt the software.<sup>10</sup> While I acknowledge these concerns, I believe they may be less important for inference in my setting for several reasons. In contrast to some prior work on open source project success that focuses on cross sectional variance in project characteristics such as license choice, my focus is on within-project variance over time due to IPR enforcement actions. That is, I include a complete set of project fixed effects to address average differences in the number of downloads across projects. Second, I treat deviance of downloads from adoption as an error in my dependent variable that can be addressed through my use of robust standard errors (Wooldridge 2002); this will affect the consistency of my parameter estimates only if this deviance changes systematically over time in a way that is correlated with my treatment.

*Developer Activity.* As noted above, my interest is in identifying the effects of IPR enforcement on the costs and benefits to developers of making contributions to an OSS project. I use the number of closed artifacts for a project in a month to measure this variable. Closed artifacts measure how successfully developers complete requests for new features, patches, and bug fixes (Chengalur-Smith et al. 2010). This measure is

---

<sup>10</sup> I also note that I do not distinguish between adopters who are first-time adopters of OSS and those who are existing users who are adopting a new version that may be available. I believe that the mechanism I describe will influence adoption by both of these groups.

consistent with past work that uses task completion to measure the amount of effort exerted by developers on a project (Herbsleb and Mockus 2003, Stewart and Gosain 2006). While developer activity on a project can be measured in many different ways (Comino et al. 2007)—such as number of bugs fixed, CVS commits, or files released to the community—I believe that using closed artifacts is more appropriate in my setting for several reasons. First, new file releases happen infrequently, so this variable will vary little month-to-month.<sup>11</sup> Second, I do not have monthly data on the number of CVS commits during the sample used for the *FireStar/DataTern v. Red Hat* case. Third, compared with other measures of developer activity, closed artifacts captures how developers respond to OSS user requests (Grewal et al. 2006). Because one major argument in my hypothesis development is that developer activity is influenced by IPR enforcement through the reduced user interest, I believe using closed artifacts provide the best lens for me to test this hypothesis.

#### 2.4.2.2 Independent Variables

*IPR Enforcement.* I construct a dummy variable labeled lawsuit that indicates whether an IPR infringement lawsuit has been filed. For *SCO v. IBM*, it is equal to 1 for observations between March 2003 and July 2003 and 0 otherwise. For *FireStar/DataTern v. Red Hat*, it is equal to 1 for observations between July 2006 and November 2006 and 0 otherwise.

*Overlap between OSS and the Focal Litigated Technology.* This variable indicates whether the OSS project overlap with the technology of the focal litigated project. For *SCO v. IBM*, SCO asserted that IBM was misappropriating and wrongly incorporating its copyrighted Unix code into Linux (SCO 2003). Therefore, the technology of the focal

---

<sup>11</sup> During the sample new files are released every seven to nine months.

litigated project in *SCO v. IBM* is the Linux operating system. The central part of the Linux operating system is called the Linux kernel, whose role is to give programs access to resources such as hard disk storage and random access memory (Hertel et al. 2003). Although a variety of distributors such as IBM and Red Hat integrate the Linux kernel into their own products to provide enhanced functions, the Linux kernel is an essential part of any type of Linux distribution and contains the potentially infringing code targeted by SCO. As noted by Al Gillen, vice president of system software research at IDC in Framingham, Mass, “SCO feels that the offending code is now so interspersed with the 2.4 and 2.5 [Linux] kernels that it will be impossible to effectively remove it. They believe the only way for it to be rectified is to go back to the 2.2 kernel and start all over again from there, and that is never going to happen” (McMillan 2003). Thus, I believe Linux kernel projects available on SourceForge will overlap with the technology of the litigated project. I generate a dummy variable “Linux\_kernel” that is equal to 1 for Linux kernel projects and 0 otherwise. Details on how I identify Linux kernel projects are included in Appendix A.

For *FireStar/DataTern v. Red Hat*, the allegedly infringed patent (No. 6,101,502) describes a method of interfacing an object-oriented software application with a relational database. The focal litigated technology is in Red Hat’s JBoss suite—particularly Hibernate 3.0’s object-relational mapping technology, which concerns a model for “employing a relational database with object oriented software” (Sanders 2006). I construct a variable “object\_relational” that is equal to one for projects that provide object-relational mapping technology. Additional details are provided in Appendix A.

*Business Projects Specific to the Focal Litigated Platform.* This variable indicates whether the OSS project is intended for a business audience and is specific to a litigated platform. For *SCO v. IBM*, the platform in question is the Linux operating system. As noted in the theory development preceding H2, the role of platform-specific investments is most significant for firms, so I focus here primarily on business-related OSS projects

that are used in conjunction with Linux. That is to say, since the unavailability of the Linux operating system would significantly decrease the value of organizational applications running on Linux, I identify the treatment group called “business\_app\_on\_Linux” as the set of business applications running on Linux. The importance of such software has been growing over time; for example, one recent report estimates that total spending in support of business application software running on Linux exceeded \$2 billion in 2008 (Gillen 2009). Further, business applications processing has accounted for an increasing share of Linux operating environment workload: by one estimate, enterprise applications such as ERP and CRM accounted for a 9% share of Linux operating environment workload in 2007 (compared with a 3.3% share in 2000) (Gillen 2009). More details on how to construct this variable are provided in Appendix A.

For *FireStar/DataTern v. Red Hat*, as noted above, JBoss software forms a platform on which other parties can build enhancements. Thus, my treatment group is the set of projects that are developed specifically for running on JBoss. Meanwhile, given that JBoss itself is intended for enterprise, I believe the intended audience for projects specific to JBoss is also firms. I further confirm this conjecture through close examination of each project’s descriptionllow prior literature in using downloads as a market-based measure of popularity

#### 2.4.2.3 Control Variables

*Time-invariant controls.* I include controls for preexisting installed based and project license type, two variables that have been shown in prior work to shape OSS project success (e.g. Subramaniam et al. 2009). My use of project-level fixed effects will cause these variables to be dropped in any regression as they are perfectly collinear with the fixed effects (Wooldridge 2002). As a result, I set these variables to zero before the IPR enforcement lawsuits; coefficients on these variables should be interpreted as

capturing how changes in the mean growth rate of user and developer activity post-enforcement are affected by these variables.

*Preexisting Installed Base.* Adopters' reaction to IPR enforcement may differ with the installed base of the project. To control for these effects, I measure each project's preexisting installed base using its accumulated downloads on SourceForge two months before the start of the sample period. I further take the log of this variable to reduce its skewness.

*Project License Type.* As noted above, a body of literature has suggested that the restrictiveness of OSS licenses could affect user interest and developer activity. I follow the literature (e.g. Lerner and Tirole 2005b, Stewart et al. 2006, Subramaniam et al. 2009) to use a dummy (denoted as `restrictive_license`) to indicate whether an OSS project is distributed under highly restrictive license (i.e. GNU General Public License or GPL). If it is distributed under GPL, I set the value of `restrictive_license` to be 1, otherwise to be 0.

*Time-varying controls.* I also control for a variety of time-varying factors that may affect user and developer activity. Because these variables may themselves be endogenous, I run my regressions with and without them for robustness. Further, I include different sets of controls in the user interest and developer activity regressions to mitigate concerns that some controls may be endogenous in some regressions. For example, I do not control for CVS commits in the developer activity regressions since common unobserved factors are likely to drive both CVS commits and closed artifacts.

*Time-Varying Controls Related to User Interest.* It is possible that the incidence of my treatments may be correlated with some other time-varying omitted factors that may also be correlated with changes in downloads over time. One such omitted variable could be time-varying improvements in the quality of the software. To address this concern, I control for the number of developers, as increases in the number of software developers may be correlated with quality of software (Mishra et al. 2002). I also add the monthly CVS commits (denoted as `cvs_commits`) for the *SCO v. IBM* case as an indicator

of successful technical refinements in each month, since a “commit occurs when a developer uploads the altered source code file, which reflects meaningful contributions to the source code” (Grewal et al. 2006). Unfortunately, as mentioned above, data on the `cvs_commits` for the complete sample period defined by *FireStar/DataTern v. Red Hat* case is unavailable to us. Therefore, I add the number of new versions released every month (denoted as `new_files`) as an alternative control for this case.

*Time-varying Controls Related to Developer Activity.* As suggested by Crowston et al. (2003), lead users’ involvement in discussions and bug reporting could affect project success. Thus, I add the number of messages posted for each project in each month as a time-varying control.

## 2.5 Empirical Models And Results

### 2.5.1 Baseline Analyses For User Interest

To test my two hypotheses for user interest, I update equation (2.1) using the treatment, treatment groups, and regression controls defined above.<sup>12</sup> I tested the impacts of the two cases of IPR enforcement (*SCO v. IBM* and *FireStar/DataTern v. Red Hat*) in separate analyses (rather than in one pooled regression) for two reasons. First, because of data constraints, I have missing observations for several months between 2003 (when the *SCO v. IBM* case was filed) and 2006 (when the *FireStar/DataTern v. Red Hat* case was filed). Thus, it is difficult to combine the two samples and jointly test the impacts of the two cases using one empirical model. Further, my approach of using separate regressions for each IPR enforcement action allows for additional flexibility in how each of my independent variables influence downloads. That is, my estimation strategy does not

---

<sup>12</sup> I also tried to test each hypothesis separately (i.e. impacts on each treated group) and the results are qualitatively similar. Due to the limited space, I only present the empirical specifications testing H1 and H2 together.

impose additional constraints on how variables like *installed\_base*, *new\_files*, and *cvs\_commits* influence downloads. The main specifications are as follows.

$$\begin{aligned}
 \text{SCO v. IBM: } \log(\text{downloads}_{it}) = & \alpha + \beta_0 \text{lawsuit}_t + \beta_1 \text{lawsuit}_t * \text{Linux\_kernel}_i + \beta_2 \\
 & \text{lawsuit}_t * \text{business\_app\_on\_Linux}_i + \gamma_1 \text{installed\_base}_{it} + \gamma_2 \\
 & \text{restrictive\_license}_{it} + \gamma_3 \text{cvs\_commits}_{it} + \gamma_4 \text{developers}_{it} + \gamma_5 \text{year2003}_t + \gamma_6 \text{month-of-year}_t \\
 & + v_i + \varepsilon_{it}
 \end{aligned} \tag{2.2}$$

$$\begin{aligned}
 \text{FireStar/DataTern v. Red Hat: } \log(\text{downloads}_{it}) = & \alpha + \beta_0 \text{lawsuit}_t + \beta_1 \text{lawsuit}_t * \\
 & \text{object\_relational}_i + \beta_2 \text{lawsuit}_t * \text{JBoss\_related}_i + \gamma_1 \text{installed\_base}_{it} + \gamma_2 \\
 & \text{restrictive\_license}_{it} + \gamma_3 \text{new\_files}_{it} + \gamma_4 \text{developers}_{it} + v_i + \varepsilon_{it}
 \end{aligned} \tag{2.3}$$

There are several things to note about equations (2.2) and (2.3). First, I employ fixed effects models in all of my analyses; these fixed effects will control for time-invariant differences (i.e.  $v_i$ ) in the average number of downloads across projects. Second, I use robust standard errors in all regressions, clustered at the project level, as recommended by Bertrand et al. (2004). Third, for *SCO v. IBM*, I include a year dummy (i.e. *year2003*) and 11 month-of-year dummies (i.e., February, ..., December) to control for time and seasonality trends in adoption. For *FireStar/DataTern v. Red Hat*, I do not include these time effects because of insufficient variance in my data source to obtain identification of these parameters.<sup>13</sup> However, in robustness checks for both equations (2) and (3), I employ month dummies that incorporate both year and month (e.g., February 2002, March 2002, ..., July 2003 for *SCO v. IBM*; August 2005, September 2005, ..., November 2006 for *FireStar/DataTern v. Red Hat*); these models will allow me to better control for unobserved time-varying factors in both models. However, in these models I am unable to identify the parameter  $\beta_0$  as it is perfectly collinear with these month dummies. Fourth, I use the log transformation of downloads as the dependent

---

<sup>13</sup> Specifically, my sample is from July 2005 to November 2006. Year dummies and month-of-year dummies are collinear with *lawsuit*.

variable because the distribution of downloads is highly skewed. Last, I do not include the direct effects of the variables *Linux\_kernel<sub>i</sub>*, *business\_app\_on\_Linux<sub>i</sub>*, *object\_relational<sub>i</sub>*, and *JBoss\_related<sub>i</sub>* as they will be absorbed in the fixed effects (i.e., since they do not vary over time they are perfectly collinear with the fixed effects). As in all difference-in-difference regressions, the identifying assumption for my model is that the average change in unobserved factors in my sample is similar for the treatment and control group (Angrist and Pischke 2009). To explore the likely accuracy of this assumption, I explore the robustness of my results to different controls, different control groups, and to a falsification exercise.

My interest is in testing whether  $\beta_1 < 0$  (H1) and  $\beta_2 < 0$  (H2) for both specification (2.2) and (2.3). The baseline coefficients for specifications (2.2) and (2.3) are shown in columns 1 through 3 in tables 2.3 and 2.4. To test the robustness of my results, I add the above set of controls incrementally. The estimated  $\beta_1$  is significantly negative in both tables 2.3 and 2.4, together confirming H1. More specifically, in the months following the filing of *SCO v. IBM*, Linux kernel projects had a 15%-16% greater decline in downloads than projects in the control group; in the same manner, in the months following the filing of *FireStar/DataTern v. Red Hat*, projects about the mapping method between an object model and a relational database were faced with an 13% greater decline relative to projects in the control group. Second, the estimated  $\beta_2$  is also significantly negative in both tables 2.3 and 2.4, together confirming H2. In the months following the filing of *SCO v. IBM*, OSS projects that were intended primarily for business organizations and that run exclusively on Linux were associated with a 34%-35% greater decline than projects in the control group. Further, in the months following the filing of *FireStar/DataTern v. Red Hat*, OSS projects specific to JBoss were faced with a 16%-17% greater decline than the control group.

To mitigate concerns about time-varying omitted variables bias, I conduct a series of robustness tests. First, as noted above, I use another (larger) set of month dummies to

control for any time-varying factors that may influence downloads (i.e., I use 18 month dummies for *SCO v. IBM* and 16 month dummies for *FireStar/DataTern v. Red Hat*). As noted above, the drawback of this strategy is that the direct effect of  $lawsuit_t$  is absorbed into these dummies and thus  $\beta_0$  is dropped. As shown in columns 4 to 6 in tables 2.3 and 2.4, the estimated coefficients (and implied marginal effects) from these models are similar to those in the main specification.

In my second robustness check I implement a falsification test to provide further evidence that my results do not reflect the presence of time-varying unobserved factors that may be correlated with lawsuit and the treated groups.<sup>14</sup> For example, one potential concern is that there may exist unobserved events that disproportionately affect my treatment group. For example, on April 10, 2006 Red Hat acquired JBoss (two months from the June 26, 2006 date of the *FireStar/DataTern v. Red Hat* case); this acquisition might have influenced user perceptions of the value of adopting projects running on JBoss. Falsification tests of this nature are commonly employed in difference-in-difference estimation (Bertrand et al. 2004). I create a dummy *false\_lawsuit* that is set equal to 1 if the observation is from the period three months before the actual filing date of lawsuit and set it to be 0 otherwise. The logic for this falsification test is that if my estimates reflect causal relationships, then the incidence of the lawsuit should have no effect on downloads prior to when the lawsuit was actually filed. That is, the interactions of *false\_lawsuit* with *Linux\_kernel* and *business\_app\_on\_Linux* should not be significantly negative.

The results are presented from column 7 to 9 in tables 2.3 and 2.4. The estimated  $\beta_1$  and  $\beta_2$  again confirm my hypotheses. Further, the coefficients on the *false\_lawsuit*

---

<sup>14</sup> To further address any concern about the serial correlation of downloads within project over time, I follow Bertrand et al. (2004) and implement another robustness test: I average the data before and after the lawsuit and run the main specifications on these averaged time-varying variables in a panel of length 2. This approach has yielded qualitatively similar results.

interactions are either insignificant or positive, which is consistent with my assertion that my results reflect a causal relationship between an IPR enforcement action and downloads for the treated projects, rather than omitted time-varying factors.<sup>15</sup>

---

<sup>15</sup> To further study the implications of the JBoss acquisition, I also experimented with using a *false\_lawsuit* that is set equal to 1 if the observation is from the period two months before the actual filing date of lawsuit. The results are qualitatively similar.

**Table 2.3 User Interest, *SCO* v. *IBM***

Dependent var: log (downloads)	Main Specification			Monthly Time Dummies			Falsification Test		
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
lawsuit	-0.484 (.036)***	-0.482 (.036)***	-0.519 (.040)***				-0.483 (.036)***	-0.480 (.036)***	-0.518 (.040)***
lawsuit · Linux_kernel	-0.161 (.083)*	-0.166 (.082)*	-0.175 (.082)*	-0.162 (.083)*	-0.166 (.082)*	-0.175 (.082)*	-0.130 (.090)+	-0.132 (.088)+	-0.142 (.089)+
lawsuit · business_app_on_Linux	-0.435 (.146)***	-0.422 (.149)***	-0.439 (.150)**	-0.436 (.146)***	-0.422 (.150)**	-0.439 (.150)**	-0.479 (.157)***	-0.462 (.161)**	-0.479 (.161)**
False_lawsuit · Linux_kernel							.149 (.071)*	.156 (.070)*	.156 (.070)*
False_lawsuit · business_app_on_Linux							-.198 (.122)	-.188 (.124)	-.188 (.124)
Controls	Project, Month of Year, Year	Project, Month of Year, Year	Project, Month of Year, Year	Project, Month	Project, Month	Project, Month	Project, Month of Year, Year	Project, Month of Year, Year	Project, Month of Year, Year
Time-varying controls	cvs commits	cvs commits, developers	cvs commits, developers	cvs commits	cvs commits, developers	cvs commits, developers	cvs commits	cvs commits, developers	cvs commits, developers
Other controls	installed base	installed base	installed base, license	installed base	installed base	installed base, license	installed base	installed base	installed base, license
Number of groups	3928	3928	3928	3928	3928	3928	3928	3928	3928
Number of obs	74632	74632	74632	74632	74632	74632	74632	74632	74632
R square	.871	.873	.873	.878	.879	.879	.871	.873	.873

Notes: 1) +: significant at 10%; \*: significant at 5%; \*\*: significant at 1%; \*\*\*: significant at 0.1%; 2) heteroskedasticity robust standard errors clustered over projects are in parentheses; 3) sample is composed of projects that have positive downloads for each month during the sampling period; 4) R-squared includes fixed effects in R-squared computation

**Table 2.4 User Interest, *FireStar/DataTern v. Red Hat***

Dependent var: log (downloads)	Main Specification			Monthly Time Dummies			Falsification Test		
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
lawsuit	.534 (.013)***	.530 (.013)***	.556 (.014)***				.534 (.013)***	.556 (.014)***	.556 (.013)***
lawsuit · object_relational	-.142 (.077)+	-.151 (.078)*	-.139 (.078)+	-.142 (.077)+	-.149 (.078)+	-.138 (.078)+	-.148 (.089)+	-.159 (.091)+	-.147 (.091)+
lawsuit · JBoss_related	-.184 (.075)**	-.181 (.076)*	-.180 (.075)*	-.184 (.076)**	-.181 (.076)*	-.181 (.075)*	-.199 (.084)**	-.197 (.084)*	-.197 (.083)*
False_lawsuit · object_relational							.023 (.090)	-.033 (.091)	-.032 (.091)
False_lawsuit · JBoss_related							-.060 (.063)	-.066 (.062)	-.066 (.062)
Controls	Project	Project	Project	Project, Month	Project, Month	Project, Month	Project	Project	Project
Time-varying controls	new files	new files, developers	new files, developers	new files	new files, developers	new files, developers	new files	new files, developers	new files, developers
Other controls	installed base	installed base	installed base, license	installed base	installed base	installed base, license	installed base	installed base	installed base, license
Number of groups	24301	24301	24301	24301	24301	24301	24301	24301	24301
Number of obs	413117	413117	413117	413117	413117	413117	413117	413117	413117
R square	.903	.903	.903	.910	.910	.910	.903	.903	.903

Notes: 1) +: significant at 10%; \*: significant at 5%; \*\*: significant at 1%; \*\*\*: significant at 0.1%; 2) heteroskedasticity robust standard errors clustered over projects are in parentheses; 3) sample is composed of projects that have positive downloads for each month during the sampling period; 4) R-squared includes fixed effects in R-squared computation.

### **2.5.2 Robustness Checks For User Interest Regressions**

In this section I describe the results of additional robustness checks of my analyses on how IPR enforcement affects user interest. One possibility is that because OSS projects with a large installed base may provide a more inviting target to IPR rights holders and subsequently exhibit greater litigation risks, it is possible that my empirical results are driven by the change in adoption from only OSS projects with the largest installed base. To examine this possibility, I construct an alternative sample that excludes OSS projects with preexisting installed base in the top 5% from the baseline sample. As shown in the table 2.5 and 2.6, my results are robust to this alternative sample.

**Table 2.5 User Interest (Alternative Sample), *SCO* v. *IBM***

Dependent var: log (downloads)	Main Specification			Monthly Time Dummies			Falsification Test		
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
lawsuit	-.441 (.038)***	-.443 (.038)***	-.472 (.042)***				-.439 (.038)***	-.441 (.038)***	-.471 (.042)***
lawsuit · Linux_kernel	-.173 (.084)*	-.179 (.083)*	-.187 (.083)*	-.173 (.084)*	-.179 (.083)*	-.187 (.083)*	-.140 (.091)+	-.145 (.090)+	-.153 (.090)+
lawsuit · business_app_on_Linux	-.432 (.149)***	-.419 (.152)**	-.432 (.152)**	-.432 (.149)***	-.419 (.152)**	-.432 (.152)**	-.475 (.160)***	-.459 (.163)**	-.472 (.163)**
False_lawsuit · Linux_kernel							.152 (.074)*	.159 (.072)*	.159 (.072)*
False_lawsuit · business_app_on_Linux							-.198 (.123)	-.188 (.124)	-.188 (.124)
Controls	Project, Month of Year, Year	Project, Month of Year, Year	Project, Month of Year, Year	Project, Month	Project, Month	Project, Month	Project, Month of Year, Year	Project, Month of Year, Year	Project, Month of Year, Year
Time-varying controls	cvs commits	cvs commits, developers	cvs commits, developers	cvs commits	cvs commits, developers	cvs commits, developers	cvs commits	cvs commits, developers	cvs commits, developers
Other controls	installed base	installed base	installed base, license	installed base	installed base	installed base, license	installed base	installed base	installed base, license
Number of groups	3730	3730	3730	3730	3730	3730	3730	3730	3730
Number of obs	70870	70870	70870	70870	70870	70870	70870	70870	70870
R square	.843	.845	.845	.852	.853	.853	.843	.845	.845

Notes: 1) +: significant at 10%; \*: significant at 5%; \*\*: significant at 1%; \*\*\*: significant at 0.1%; 2) heteroskedasticity robust standard errors clustered over projects are in parentheses; 3) the alternative sample is the baseline sample excluding projects having top 5% preexisting installed base; 4) R-squared includes fixed effects in R-squared computation

**Table 2.6 User Interest (Alternative Sample), *FireStar/DataTern v. Red Hat***

Dependent var: log (downloads)	Main Specification			Monthly Time Dummies			Falsification Test		
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
lawsuit	.636 (.014)***	.631 (.014)***	.645 (.014)***				.636 (.014)***	.631 (.014)***	.645 (.014)***
lawsuit · object_relational	-.136 (.079)+	-.143 (.079)+	-.134 (.079)+	-.135 (.079)+	-.141 (.079)+	-.132 (.079)+	-.142 (.091)+	-.153 (.092)+	-.144 (.092)+
lawsuit · JBoss_related	-.190 (.076)**	-.186 (.077)*	-.190 (.076)*	-.190 (.076)**	-.187 (.077)*	-.191 (.075)**	-.204 (.085)**	-.202 (.085)*	-.206 (.084)*
False_lawsuit · object_relational							-.028 (.092)	-.038 (.093)	-.037 (.093)
False_lawsuit · JBoss_related							-.055 (.066)	-.061 (.065)	-.061 (.065)
Controls	Project	Project	Project	Project, Month	Project, Month	Project, Month	Project	Project	Project
Time-varying controls	new files	new files, developers	new files, developers	new files	new files, developers	new files, developers	new files	new files , developers	new files , developers
Other controls	installed base	installed base	installed base, license	installed base	installed base	installed base, license	installed base	installed base	installed base, license
Number of groups	23089	23089	23089	23089	23089	23089	23089	23089	23089
Number of obs	392513	392513	392513	392513	392513	392513	392513	392513	392513
R square	.868	.868	.868	.878	.878	.879	.868	.868	.868

Notes: 1) +: significant at 10%; \*: significant at 5%; \*\*: significant at 1%; \*\*\*: significant at 0.1%; 2) heteroskedasticity robust standard errors clustered over projects are in parentheses; 3) the alternative sample is the baseline sample excluding projects with top 5% preexisting installed base; 4) R-squared includes fixed effects in R-squared computation.

As noted above, a key requirement in my difference-in-difference approach is that unmeasured factors affect the treatment group and control group equally in my regressions. To further probe the validity of this assumption, I conduct a set of subsample analyses in which I provide a more precisely matched control group for the treatment groups used to test H2. Unfortunately, due to the nature of the treatment group for H1, it is very difficult to similarly find a more precisely matched control group for these projects. Thus, in these analyses I am only able to examine the robustness of my results for H2.

I compile a new set of narrower control groups in my tests of H2 for both cases. For *SCO v. IBM*, I identify the new control group as the set of projects that are intended for business users and can run on operating systems other than Linux. For *FireStar/DataTern v. Red Hat*, I construct the new control group as the set of projects that are related to Java Platform, Enterprise Edition, but not related to JBoss. I believe that these control groups are closely matched to the treatment groups, so without any lawsuit, the treatment group should exhibit the same time trend in downloads as this matched control group. As show in table 2.7, my results are robust to this alternative sample.

**Table 2.7 User Interest, Sub-Sample**

<b><i>SCO v. IBM</i></b>						
Dependent var: log (downloads)	Main Specification			Monthly Time Dummies		
	(1)	(2)	(3)	(4)	(5)	(6)
lawsuit	-.588 (.214)**	-.583 (.214)**	-.599 (.256)*			
lawsuit · business_app_on_Linux	-.311 (.165)+	-.296 (.166)+	-.306 (.173)+	-.310 (.164)+	-.296 (.167)+	-.306 (.174)+
Controls	Project, Month of Year, Year	Project, Month of Year, Year	Project, Month of Year, Year	Project, Month	Project, Month	Project, Month
Time-varying controls	cvs commits	cvs commits, developers	cvs commits, developers	cvs commits	cvs commits, developers	cvs commits, developers
Other controls	installed base	installed base	installed base, license	installed base	installed base	installed base, license
Number of groups	93	93	93	93	93	93
Number of obs	1767	1767	1767	1767	1767	1767
R square	.852	.852	.852	.862	.863	.863
<b><i>FireStar/DataTern v. Red Hat</i></b>						
Dependent var: log (downloads)	Main Specification			Monthly Time Dummies		
	(1)	(2)	(3)	(4)	(5)	(6)
lawsuit	.339 (.153)*	.345 (.149)*	.349 (.149)*			
lawsuit · JBoss_related	-.130 (.083)+	-.117 (.083)+	-.122 (.083)+	-.130 (.084)+	-.117 (.083)+	-.121 (.084)+
Controls	Project	Project	Project	Project, Month	Project, Month	Project, Month
Time-varying controls	new files	new files, developers	new files, developers	new files	new files, developers	new files, developers
Other controls	installed base	installed base	installed base, license	installed base	installed base	installed base, license
Number of groups	168	168	168	168	168	168
Number of obs	2856	2856	2856	2856	2856	2856
R square	.919	.920	.920	.925	.926	.926

Notes: 1) +: significant at 10%; \*: significant at 5%; \*\*: significant at 1%; \*\*\*: significant at 0.1%; 2) heteroskedasticity robust standard errors clustered over projects are in parentheses; 3) R-squared includes fixed effects in R-squared computation.

One potential question in the construction of H2 is whether downloads for all types of platform-specific OSS projects decline in the face of an IPR enforcement action against a platform (instead of just those business-intended projects). This question is particularly an issue for the *SCO v. IBM* case, since there are a large number of projects that use the Linux platform but that are not intended for business use. To investigate this question, I compiled for the *SCO v. IBM* case a new treatment group composed of all applications running exclusively on Linux. However, I did not find a significant difference between the changes in downloads for this group of projects and other projects. This further confirms my hypothesis that IPR enforcement particularly affects user interest toward those business-intended OSS projects specific to a litigated OSS platform.

### **2.5.3 Baseline Analyses For Developer Activity**

The dependent variable to test hypotheses 3 and 4 is the number of closed artifacts. The summary statistics in table 1 show that the mean number of closed artifacts is either 5.4 (for *SCO v. IBM*) or 3.6 (for *FireStar/DataTern v. Red Hat*); in these types of settings, count data models are appropriate (Chapter 20, Cameron and Trivedi 2005).<sup>16</sup> Therefore, I employ count data models and use conditional fixed effects to control for differences in developer activity across OSS projects. Given the amount of time to complete a programming task, I expect a delay between the time when work on an artifact is begun and when completed tasks are reported. Accordingly, if an IPR enforcement action affects developer incentives, I may expect some delay between when the enforcement action is initiated and when this change in behavior is reflected in the

---

<sup>16</sup> However, as a robustness check, I also implemented a linear model. The results are consistent with my baseline count data model.

number of closed artifacts. Therefore, in my empirical models, I lag all independent variables by one month.<sup>17</sup> More specifically, suppose the number of closed artifacts for OSS project  $i$  in month  $t$  (denoted as  $\text{Closed\_Artifacts}_{it}$ ) follows a Poisson process with parameter  $\lambda_{it}$  taking the form  $\lambda_{it} = \exp(\mathbf{X}_{it-1}'\beta)$ . Also suppose  $\alpha_i$  is a project-specific and time-invariant variable that incorporates unobserved heterogeneity of the project  $i$ . So  $E(\text{Closed\_Artifacts}_{it}|\mathbf{X}_{it-1},\alpha_i)=\lambda_{it}=\alpha_i\exp(\mathbf{X}_{it-1}'\beta)$ , where

$$\begin{aligned} \text{SCO v. IBM: } \mathbf{X}_{it-1}'\beta = & \beta_0 \text{lawsuit}_{t-1} + \beta_1 \text{lawsuit}_{t-1} * \text{Linux\_kernel}_i + \\ & \beta_2 \text{lawsuit}_{t-1} * \text{business\_app\_on\_Linux}_i + \gamma_1 \text{installed\_base}_{it-1} \\ & + \gamma_2 \text{restrictive\_license}_{it-1} + \gamma_3 \text{msg\_posted}_{it-1} + \gamma_4 \text{year2003}_{t-1} + \gamma_5 \text{month-of-year}_{t-1} + \varepsilon_{it-1} \end{aligned} \quad (2.4)$$

$$\begin{aligned} \text{FireStar/DataTern v. Red Hat: } \mathbf{X}_{it-1}'\beta = & \beta_0 \text{lawsuit}_{t-1} + \\ & \beta_1 \text{lawsuit}_{t-1} * \text{object\_relational}_i + \beta_2 \text{lawsuit}_{t-1} * \text{JBoss\_related}_i \\ & + \gamma_1 \text{installed\_base}_{it-1} + \gamma_2 \text{restrictive\_license}_{it-1} + \gamma_3 \text{msg\_posted}_{it-1} + \varepsilon_{it-1} \end{aligned} \quad (2.5)$$

I estimate these models using maximum likelihood with robust standard errors clustered at the project level. Similar to the specifications related to user interest, for *SCO v. IBM*, I include a year dummy (i.e. year2003) and 11 month-of-year dummies (i.e., February, ..., December) to control for time and seasonality trends in developer activity. For *FireStar/DataTern v. Red Hat*, I am not able to identify these time effects because of insufficient variance in my data. Second, as before, I set *installed\_base<sub>it</sub>* and *restrictive\_license<sub>it</sub>* to zero before the IPR enforcement lawsuits to prevent them from being differenced out of the regression; I add controls incrementally to explore the robustness of my results. Third, I also implement robustness checks for both equations (4) and (5) by employing another (larger) set of month dummies to control for any time-varying factors that may influence closed artifacts (i.e., I use 17 month dummies for *SCO*

---

<sup>17</sup> I also tried to lag independent variables by two months. The results are very consistent with results based on one-month lag.

*v. IBM* and 15 month dummies for *FireStar/DataTern v. Red Hat*). Last, the direct effects of the variables  $Linux\_kernel_i$ ,  $business\_app\_on\_Linux_i$ ,  $object\_relational_i$ , and  $JBoss\_related_i$  are absorbed in the parameter  $\alpha_i$  and not included in the specifications.

My interest is testing whether  $\beta_1 < 0$  (H3) and  $\beta_2 < 0$  (H4) for specification (2.4) and (2.5). The estimated coefficients are shown in columns 1 to 3 in tables 2.8 and 2.9. First, the estimated  $\beta_1$  is significantly negative for both cases, together confirming H1: the parameter estimates imply that Linux kernel projects experienced 45%-47% fewer closed artifacts than the control group in the months following the filing of *SCO v. IBM*; similarly, in the months following the filing of *FireStar/DataTern v. Red Hat*, projects about the mapping method between an object model and a relational database had 89%-90% fewer closed artifacts than the control group. Second, the estimated  $\beta_2$  is also significantly negative for both cases, together confirming H4. The parameter estimates imply that in the months following the filing of *SCO v. IBM*, OSS projects that were intended primarily for business organizations and that run exclusively on Linux had 86%-87% fewer closed artifacts than the control group; meanwhile, in the months following the filing of *FireStar/DataTern v. Red Hat*, OSS projects specific to the JBoss suite also experienced about 84% fewer closed artifact than the control group. Moreover, as shown in column 4 to column 6 in tables 2.8 and 2.9, the results based on the specifications with a larger set of month dummies are consistent with the baseline results.

As I had done in my analyses on user interest, I conduct a falsification test to provide further evidence that the results do not reflect the presence of time-varying unobserved factors that may be correlated with lawsuit and the treated groups. The results are presented in columns 7 to 9 in tables 2.8 and 2.9. The estimated  $\beta_1$  and  $\beta_2$  again confirm my hypotheses. Further, all of the coefficients for the interactions with *false\_lawsuit* are statistically insignificant, offering additional evidence that my results reflect the effect of IPR enforcement on developer activity instead of some omitted time-varying factors.

**Table 2.8 Developer Activity, *SCO* v. *IBM***

closed artifacts	Main Specification			Monthly Time Dummies			Falsification Test		
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
Lawsuit	.244 (.363)	.187 (.334)	.194 (.396)				.243 (.363)	.186 (.335)	.193 (.396)
lawsuit · Linux kernel	-.638 (.192)***	-.603 (.194)**	-.602 (.196)**	-.638 (.193)***	-.603 (.194)**	-.603 (.196)**	-.681 (.198)***	-.632 (.202)**	-.632 (.204)**
lawsuit · business_app_on_Linux	-2.070 (.840)**	-2.009 (.828)*	-2.006 (.828)*	-2.070 (.840)*	-2.008 (.828)*	-2.006 (.828)*	-2.107 (1.030)*	-2.035 (1.015)*	-2.031 (1.016)*
False_lawsuit · Linux kernel							-.214 (.139)	-.152 (.148)	-.152 (.148)
False_lawsuit · business_app_on_Linux							-.186 (1.031)	-.130 (1.015)	-.130 (1.015)
Controls	Project, Month of Year, Year	Project, Month of Year, Year	Project, Month of Year, Year	Project, Month	Project, Month	Project, Month	Project, Month of Year, Year	Project, Month of Year, Year	Project, Month of Year, Year
Time-varying controls		msg posted	msg posted		msg posted	msg posted		msg posted	msg posted
Other controls	installed base	installed base	installed base, license	installed base	installed base	installed base, license	installed base	installed base	installed base, license
Number of groups	904	904	904	904	904	904	904	904	904
Number of obs	16272	16272	16272	16272	16272	16272	16272	16272	16272
Log pseudolikelihood	-57559.54	-56910.27	-56910.08	-57525.69	-56873.20	-56873.01	-57556.71	-56908.88	-56908.68

Notes: 1) +: significant at 10%; \*: significant at 5%; \*\*: significant at 1%; \*\*\*: significant at 0.1%; 2) robust standard errors clustered over projects are in parentheses; 3) sample is composed of projects that have total number of closed artifacts above the 25th percentile of closed artifacts (i.e. above 3). 4) The above 16,272 observations are from 904 projects over 18 months (one month is dropped because I use the lagged closed artifacts).

**Table 2.9 Developer Activity, *FireStar/DataTern v. Red Hat***

closed artifacts	Main Specification			Monthly Time Dummies			Falsification Test		
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
lawsuit	-.391 (.177)*	-.413 (.176)*	-.460 (.180)**				-.391 (.177)*	-.413 (.176)*	-.460 (.179)**
lawsuit · Object-Relational	-2.303 (1.039)*	-2.296 (1.039)*	-2.242 (1.033)*	-2.303 (1.040)*	-2.295 (1.038)*	-2.241 (1.033)*	-2.477 (1.133)*	-2.471 (1.131)*	-2.417 (1.125)*
lawsuit · JBoss	-1.842 (.464)***	-1.868 (.463)***	-1.882 (.466)***	-1.842 (.464)***	-1.871 (.463)***	-1.884 (.466)***	-1.976 (.468)***	-2.012 (.464)***	-2.026 (.465)***
False_lawsuit · Object-Relational							-1.022 (.930)	-1.025 (.927)	-1.025 (.927)
False_lawsuit · JBoss							-.698 (.477)	-.736 (.452)	-.735 (.452)
Controls	Project	Project	Project	Project, Month	Project, Month	Project, Month	Project	Project	Project
Time-varying controls		msg posted	msg posted		msg posted	msg posted		msg posted	msg posted
Other controls	installed base	installed base	installed base, license	installed base	installed base	installed base, license	installed base	installed base	installed base, license
Number of groups	2311	2311	2311	2311	2311	2311	2311	2311	2311
Number of obs	36976	36976	36976	36976	36976	36976	36976	36976	36976
Log pseudolikelihood	-108080.07	-108026.76	-107989.63	-107684.41	-107625.12	-107588.30	-108062.88	-108008.53	-107971.42

Notes: 1) +: significant at 10%; \*: significant at 5%; \*\*: significant at 1%; \*\*\*: significant at 0.1%; 2) robust standard errors clustered over projects are in parentheses; 3) sample is composed of projects that have total number of closed artifacts above the 25th percentile of closed artifacts (i.e. above 2); 4) The above 36,976 observations are from 2,311 projects over 16 months (one month is dropped because I use the lagged closed artifacts).

Last, to further boost confidence in my results, I implement all the above tests for H3 and H4 using an alternative sample that consists of all projects that have total number of artifacts larger than zero (recall that my baseline sample includes only projects with total number of closed artifacts above the 25th percentile in the sample). As shown in tables 2.10 and 2.11, the results are qualitatively similar to the baseline ones. Similarly, as I had done in my robustness test of H2, I also construct a set of closely matched control groups—business projects running on other platforms for *SCO v. IBM* and projects that are related to J2EE Platform but not JBoss for *FireStar/DataTern v. Red Hat*. As shown in table 2.12, my results from the subsample analyses are consistent with the baseline results.

**Table 2.10 Developer Activity (Alternative Sample), *SCO v. IBM***

closed artifacts	Main Specification			Monthly Time Dummies			Falsification Test		
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
lawsuit	.232 (.360)	.176 (.331)	.186 (.393)				.230 (.360)	.175 (.332)	.185 (.394)
lawsuit · Linux_kernel	-.643 (.193)***	-.608 (.194)**	-.608 (.197)**	-.643 (.193)***	-.609 (.194)**	-.608 (.197)**	-.690 (.199)***	-.642 (.203)**	-.641 (.206)**
lawsuit · business_app_on_Linux	-1.406 (.748)+	-1.353 (.734)+	-1.348 (.734)+	-1.406 (.748)+	-1.352 (.734)+	-1.347 (.734)+	-1.452 (.869)+	-1.388 (.852)+	-1.383 (.852)+
False_lawsuit · Linux_kernel							-.236 (.143)	-.174 (.154)	-.174 (.154)
False_lawsuit · business_app_on_Linux							-.230 (1.018)	-.178 (1.003)	-.178 (1.003)
Controls	Project, Month of Year, Year	Project, Month of Year, Year	Project, Month of Year, Year	Project, Month	Project, Month	Project, Month	Project, Month of Year, Year	Project, Month of Year, Year	Project, Month of Year, Year
Time-varying controls		msg posted	msg posted		msg posted	msg posted		msg posted	msg posted
Other controls	installed base	installed base	installed base, license	installed base	installed base	installed base, license	installed base	installed base	installed base, license
Number of groups	1277	1277	1277	1277	1277	1277	1277	1277	1277
Number of obs	22986	22986	22986	22986	22986	22986	22986	22986	22986
Log pseudolikelihood	-59360.73	-58710.34	-58709.96	-59327.46	-58673.87	-58673.51	-59357.28	-58708.48	-58708.11

Notes: 1) +: significant at 10%; \*: significant at 5%; \*\*: significant at 1%; \*\*\*: significant at 0.1%; 2) robust standard errors clustered over projects are in parentheses; 3) the alternative sample is composed of projects that have total number of closed artifacts larger than zero; 4) The above 22,986 observations are from 1,277 projects over 18 months (one month is dropped because I use the lagged closed artifacts).

**Table 2.11 Developer Activity (Alternative Sample), *FireStar/DataTern v. Red Hat***

closed artifacts	Main Specification			Monthly Time Dummies			Falsification Test		
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
lawsuit	-0.394 (.173)*	-0.416 (.173)*	-0.462 (.176)**				-0.394 (.173)*	-0.416 (.173)*	-0.462 (.176)**
lawsuit · object_relational	-2.308 (1.038)*	-2.302 (1.037)*	-2.249 (1.032)*	-2.308 (1.038)*	-2.301 (1.036)*	-2.249 (1.032)*	-2.476 (1.133)*	-2.470 (1.131)*	-2.417 (1.125)*
lawsuit · JBoss_related	-1.753 (.458)***	-1.779 (.458)***	-1.792 (.462)***	-1.753 (.458)***	-1.782 (.458)***	-1.794 (.462)***	-1.887 (.461)***	-1.923 (.458)***	-1.936 (.459)***
False_lawsuit · object_relational							-0.957 (.901)	-0.960 (.898)	-0.960 (.898)
False_lawsuit · JBoss_related							-0.698 (.477)	-0.736 (.452)	-0.735 (.453)
Controls	Project	Project	Project	Project, Month	Project, Month	Project, Month	Project	Project	Project
Time-varying controls		msg posted	msg posted		msg posted	msg posted		msg posted	msg posted
Other controls	installed base	installed base	installed base, license	installed base	installed base	installed base, license	installed base	installed base	installed base, license
Number of groups	3239	3239	3239	3239	3239	3239	3239	3239	3239
Number of obs	51824	51824	51824	51824	51824	51824	51824	51824	51824
Log pseudolikelihood	-111485.41	-111432.58	-111397.70	-111071.94	-111012.82	-110978.25	-111469.03	-111415.18	-111380.33

Notes: 1) +: significant at 10%; \*: significant at 5%; \*\*: significant at 1%; \*\*\*: significant at 0.1%; 2) robust standard errors clustered over projects are in parentheses; 3) the alternative sample is composed of projects that have total number of closed artifacts larger than zero; 4) the above 51,824 observations are from 3,239 projects over 16 months (one month is dropped because I use the lagged closed artifacts).

**Table 2.12 Developer Activity, Sub-Sample**

<b><i>SCO v. IBM</i></b>						
closed artifacts	Main Specification			Monthly Time Dummies		
	(1)	(2)	(3)	(4)	(5)	(6)
lawsuit	-0.601 (.833)	-0.612 (.818)	.964 (1.192)			
lawsuit · business_app_on_Linux	-2.222** (.873)	-2.179** (.867)	-2.002** (.843)	-2.221** (.873)	-2.195** (.867)	-2.018* (.845)
Controls	Project, Month of Year, Year	Project, Month of Year, Year	Project, Month of Year, Year	Project, Month	Project, Month	Project, Month
Time-varying controls		msg posted	msg posted		msg posted	msg posted
Other controls	installed base	installed base	installed base, license	installed base	installed base	installed base, license
Number of groups	25	25	25	25	25	25
Number of obs	450	450	450	450	450	450
Log pseudolikelihood	-1479.428	-1477.027	-1429.667	-1448.332	-1447.304	-1400.948
<b><i>FireStar/DataTern v. Red Hat</i></b>						
closed artifacts	Main Specification			Monthly Time Dummies		
	(1)	(2)	(3)	(4)	(5)	(6)
lawsuit	-1.990+ (1.129)	-2.351* (1.098)	-1.211 (1.258)			
lawsuit · JBoss_related	-1.577+ (.994)	-1.631+ (1.005)	-.931 (1.125)	-1.577+ (.999)	-1.714+ (1.011)	-1.063 (1.084)
Controls	Project	Project	Project	Project, Month	Project, Month	Project, Month
Time-varying controls		msg posted	msg posted		msg posted	msg posted
Other controls	installed base	installed base	installed base, license	installed base	installed base	installed base, license
Number of groups	27	27	27	27	27	27
Number of obs	432	432	432	432	432	432
Log pseudolikelihood	-1202.439	-1194.043	-1185.241	-1130.489	-1105.896	-1098.621

Notes: 1) +: significant at 10%; \*: significant at 5%; \*\*: significant at 1%; 2) robust standard errors clustered over projects are in parentheses.

## 2.6 Conclusions

Motivated by a stream of literature on how the introduction of IPR affects the use and production of knowledge in the public domain, I take a first step towards investigating the implications of formal IPR for OSS use and development. Specifically, I examine how the enforcement of IPR influence two key metrics of OSS success—the success of an OSS project in attracting user adoption (which I label as user interest) and the success of an OSS project in attracting developers to contribute (which I label as developer activity). I hypothesize that IPR enforcement can affect user interest and developer activity among two key groups: those that display technology overlap with the litigated project and those business OSS projects specific to the litigated platform. To test these hypotheses, I examine the impact of two widely known and reported IPR enforcement actions—*SCO v. IBM* and *FireStar/DataTern v. Red Hat*—on downloads and closed artifacts of OSS projects extracted from SRDA. My empirical evidence strongly supports my hypotheses. The empirical results are also robust to a series of robustness checks.

This study provides several insights. First, my results advance recent research on the effects of formal IPR protection on the production and use of public knowledge goods. While these issues have been studied empirically in other settings such as scientific research (e.g., Murray and Stern 2007), they have not to my knowledge been studied in the context of OSS. This is surprising, given that the large number of recent IPR enforcement actions against OSS projects (e.g., surrounding mobile phone technology and the Android platform) suggests great salience of these issues to the OSS community.

My research also advances the literature on OSS project success. I build upon recent work that has explored how factors outside of an OSS project’s characteristics influence user and developer behavior (e.g., Chengalur-Smith et al. 2010). In particular, I

highlight the presence and enforcement of formal IPR, and how these influence the benefits to participating in the OSS community.

My research shows that IPR enforcement has a significant impact on user behavior: downloads of certain types of projects that were related to litigated ones experienced a significant decline after the announcement of an IPR enforcement action. Several firms who commercialize OSS projects, such as Red Hat and Novell, have offered warranty or indemnification programs to mitigate potential costs of legal infringement for their customers. My results suggest the potential value of such programs to users.

My research also shows that IPR enforcement actions are associated with a significant decline in developer activity. This finding complements recent work that has studied the implications of formal IPR in the proprietary software industry (e.g., Graham and Mowery 2003, Cockburn and MacGarvie 2011). In particular, while prior work has shown that the presence of formal IPR like patents can lead to a decline in new product entry from proprietary software firms (Cockburn and MacGarvie 2011), there has been no similar work in OSS. My research advances the literature by demonstrating the implications of formal IPR for new products developed as OSS.

Last, I add to recent efforts among Information Systems researchers to evaluate the impact of changes to legal regimes on economic outcomes in settings such as online file sharing (Bhattacharjee et al. 2006), the supply of creative work (Png and Wang 2009), and identity theft (Romanosky et al. 2008). In particular, I add to recent efforts studying how IPR enforcement actions influence technology adoption (Tucker 2012): I highlight the role of the IPR environment and show how IPR enforcement actions will influence the prevailing model of OSS adoption and diffusion.

While my research provides an important first step towards understanding how IPR enforcement actions influence OSS project success, there are limitations. For one, my difference-in-difference approach relies upon revealed preferences to infer how the

costs and benefits of using and developing OSS change after an IPR enforcement action. While this approach has the advantage that it does not require OSS participants to recall their costs and benefits of adoption and use, it would be valuable to corroborate my findings using measured costs and benefits, potentially from survey data. Second, my approach relies on the study of two specific events over two specific time periods. I leave it to future work to study whether my results generalize to other time periods. Third, my empirical investigation on H1 and H3 focused on lawsuits targeting against OSS platforms, so it would be valuable to test whether these hypotheses hold in other settings.

While my research takes a first step at examining the implications of formal IPR enforcement on OSS project success, there are ample opportunities for future research. First, IPR enforcement might induce longer run changes to behavior by developers and users. Second, future work might explore what institutional mechanisms might mitigate the effects of IPR enforcement on the OSS community. The third chapter of this dissertation studies whether OSS patent pools promote new OSS product entry, representing a first step in this direction. My findings are provocative, and I hope that my research motivates further empirical work into many interesting research questions in this important area.

# **CHAPTER 3**

## **PATENT POOLS, THICKETS, AND OPEN SOURCE SOFTWARE**

### **ENTRY BY START-UP FIRMS**

#### **3.1 Introduction**

There has been significant growth in the development and commercialization of open source software (OSS) in recent years. A recent survey has indicated that worldwide revenue from commercializing OSS reached \$1.8 billion in 2006 and is expected to reach \$5.8 billion in 2011 (Broersma 2007). However, one barrier to increasing growth in OSS innovation is the risk to OSS producers of infringing existing intellectual property rights (IPR) such as patents.

While patent thickets have been shown to deter firm entry into the software industry (Cockburn and MacGarvie 2011), the nature of OSS innovation suggests that innovation in this type of software may be particularly sensitive to IPR enforcement actions for several reasons. First, the distributed, incremental development approach to developing OSS implies that innovation is highly cumulative. Moreover, this same approach makes it difficult to identify the provenance of source code, imposing high costs on OSS developers who wish to identify potentially infringing technologies. Anecdotal evidence suggests that IPR enforcement against OSS can have a significant impact on software markets, as demonstrated by the well-known set of lawsuits by the SCO Group against Linux as well as the most recent set of lawsuits among Apple, Samsung, and HTC that are related to the Android open source platform.

One way to ameliorate the patent thickets problem for OSS innovation is the creation of royalty-free patent pools, a special type of patent pool that offers royalty-free

usage of patents to any firm that promises not to sue the pool’s beneficiaries (Lévêque and Ménière 2007, Serafino 2007, Hall and Helmers 2011).<sup>18</sup> However, the empirical effectiveness of such patent pools in mitigating the patent thicket problem is largely unknown. Prior studies of patent pools have focused largely on traditional patent pools that delineate specific licensing rules and restrictions to non-pool members. These traditional patent pools have been shown to have an uncertain impact on the rate of innovation (e.g., Lerner and Tirole 2004, Lampe and Moser 2009, Joshi and Nerkar 2011).

There is also reason to question the impact of royalty-free patent pools on innovation and competition. Inventors may have insufficient incentives to provide intellectual property as a public good to the community (Gambardella and Hall 2006), and so royalty-free patent pools may contain few or low-quality patents that have little impact on innovative activity. One software industry commentator, for instance, suggests that “the perception is that bigger companies only commit their least-effective, least-important patents to a patent pool” (Seeker 2010). In their recent analysis of patents pledged to the “Eco-Patent Commons,” established in 2008 to provide royalty-free access to third parties to patented technologies related to climate change, Hall and Helmers (2011) find mixed evidence on their impact on innovation in a different industry setting. While their analysis is preliminary—due to the short period of time following the establishment of this particular patent pool—the authors conclude that patents contributed to the Eco-Patent Commons have “no discernible impact on the diffusion of the knowledge embedded in the protected technologies to other patenting firms” (Hall and Helmers 2011).

---

<sup>18</sup> Hall and Helmers (2011) refer to a royalty-free patent pool as a patent commons. I have found both labels are currently in use, and use the label royalty-free patent pool both to draw a distinction to traditional patent pools and to avoid confusion with the OSS “Patent Commons” that was developed by IBM and is a focus of this study.

Motivated by these observations, I take a first step to evaluate how innovation is influenced by the creation of a royalty-free patent pool that is made available to the OSS community. Specifically, I examine the impact of such an OSS patent pool on entry by start-ups using an OSS license (which I refer to as “OSS entry”).<sup>19</sup> My focus on this margin of innovation is motivated by several considerations. First, the formation of OSS patent pools may have a particularly strong impact on start-up innovation. Unlike large firms, start-ups usually lack the R&D capabilities and financial resources required to expand their own patent portfolios, so it is difficult for them to navigate patent thickets using other approaches such as cross-licensing agreements.<sup>20</sup> This is particularly likely to be the case for start-ups that produce OSS and who, for a variety of reasons, may be unlikely to patent their innovations. Further, as has been highlighted elsewhere, my knowledge of the implications of formal IP rights for OSS innovation is still quite limited (Lerner and Tirole 2005a, von Hippel and von Krogh 2003). My focus on entry is motivated by my context: since many firms producing under an OSS license do not patent, traditional patent-based measures of innovation are inappropriate for my setting. In short, due to the uncertainty about the effectiveness of OSS patent pools, my research strategy is to examine their impact along a margin of innovation where they are more likely to matter. I leave the implications for other types of innovation (such as entry under proprietary licenses and the behavior of large firms) for future work.

To motivate my empirical analysis, I develop a model that shows how an OSS patent pool can change the bargaining game between a proprietary incumbent patent holder and a start-up firm. The model, which builds on prior work by Llobet (2003) and

---

<sup>19</sup> I similarly refer to firms that have engaged in OSS as “OSS firms.” More precisely, an OSS firm is defined as one that develops, uses, or commercializes source code that meets the Open Source Initiative definition of open source software.

<sup>20</sup> As noted by Matt Asay, the chief operating officer at Canonical (the company behind the Ubuntu Linux operating system), “this [type of patent collective] may be the only refuge for start-ups and others, like Red Hat, that don’t have an aggressive patent-acquisition policy.” (Matt Asay 2010).

Galasso and Schankerman (2010), shows that increases in the size of the pool influence the outcome of the litigation game and consequently the start-up's OSS entry decision. Comparative statics from the model show that (i) changes in the size of the OSS patent pool related to a software segment facilitates OSS entry by start-up firms into the same segment; and (ii) the marginal effect of the pool on OSS entry will be especially large in software segments where the cumulativeness of innovation is high or where patent ownership in a segment is concentrated.

Focusing on one major OSS patent pool—the Patent Commons—I examine the empirical salience of these predictions using a unique data set. I assemble data on OSS entry using data on product releases from 2,054 start-up software firms contained in the Gale database “PROMT”. Following prior work that has examined the extent to which patents deter entry into the software industry (Cockburn and MacGarvie 2009, 2011), I allocate patents to software product market segments by (i) identifying the main technological classes of patents acquired and cited by single software product producers and (ii) comparing a set of keywords from a software segment classification with the keywords from the patent's technological classes. I use this mapping to identify the number of pool patents related to each market segment, as well as the cumulativeness of innovation and patent ownership concentration in the segment.

Using count data conditional fixed effects models, my empirical strategy examines whether time series variation in the number of patent claims in the OSS patent pool related to a software segment is associated with changes in the number of OSS entrants into that segment. My initial approach is to assume that changes in the number of pool claims are uncorrelated with omitted variables that may influence OSS entry by start-ups. I later relax this assumption, employing count data models with instrumental variables using Generalized Method of Moments (GMM) estimation. My first instrument uses the pre-sampling stock of patents held by the major patent pool contributor—IBM—across each of the market segments in my sample. I interact this variable with time

dummies to obtain time series variation. My second instrument uses the number of IBM patents that were opposed at the European Patent Office in each software segment-year.

My results suggest that a 10% increase in the pool's patent claims in a software segment is associated with a 1.5%-2.9% increase in the rate of OSS entry by start-ups into that segment.<sup>21</sup> The marginal impact of the OSS patent pool is significantly greater in segments where the cumulativeness of innovation is high: a 10% increase in the pool's patent claims is associated with a 3.8%-5.6% increase in the rate of OSS entry when the cumulativeness of innovation is at its 90th percentile, compared to no significant increase when cumulativeness is at its 10th percentile. The effect of OSS patents pools is also greater when the concentration of patent ownership is high. A 10% increase in the pool's patent claims is associated with a 1.3%-1.7% increase when patent concentration is at its 90th percentile and no increase when patent concentration is at its 10th percentile, however the statistical significance of these results vary somewhat across specifications. I explore the robustness of all of my results to adding a variety of controls, as well as to GMM instrumental variables estimation. These additional analyses suggest a causal interpretation to my results.

My study provides the first large sample evidence on how the provision of a royalty-free patent pool shapes OSS entry by start-ups. As such, my research adds to the literature that looks at how IPR licensing and enforcement influences OSS innovation (Graham and Mowery 2005, Lerner and Tirole 2005b, Maurer and Scotchmer 2006) as well as recent work that studies firm decisions to commercialize innovations using an OSS license (Bonaccorsi et al. 2006; Dahlander 2007, Fosfuri et al. 2008).

My research also adds to recent studies that empirically investigate the economic implications of patent pools. While previous studies on patent pools have focused mainly

---

<sup>21</sup> All marginal effects are evaluated at mean values of covariates.

on their design (e.g., Lerner and Tirole 2004, Lerner et al. 2007, Layne-Farrar and Lerner 2010), a few recent empirical studies (Lampe and Moser 2010, Joshi and Nerkar 2011, Hall and Helmers 2011) have begun to look at the impact of patent pools on the direction of innovative activities. However, to the best of my knowledge, there has been no empirical research on how patent pools shape start-up entry.

Finally, I also contribute to the literature by examining the potential anti-commons problems from strategic patenting and the impact of patent thickets on entry into the software industry (e.g., Cockburn and MacGarvie 2011, Ziedonis 2004). While the patent thickets problem can be examined from different perspectives, I highlight the roles of cumulateness of innovation and patent ownership concentration as two different and important dimensions of patent thickets. I propose mechanisms under which these characteristics may interact with the OSS patent pool to determine start-up entry costs. Thus, my research also provides empirical evidence on the effectiveness of mechanisms meant to mitigate the anti-commons problem, such as the establishment of patent pools or standard-setting organizations (Shapiro 2001, Rysman and Simcoe 2008).

### **3.2 Theory And Hypotheses**

I study the implications of a royalty-free patent pool for entry into an OSS market segment. While I describe the institutional characteristics of these patent pools below, their key features are that licensees are neither required to pay a license fee nor are they required to contribute patents to the pool. Patent pool licensees can use the patent pool defensively when facing litigation from non-pool patent holders. In this section I develop a simple model that delineates how the size of an OSS patent pool, together with the nature of innovative activity and the distribution of patent holdings in the segment, shapes start-up decisions to enter into that segment by releasing a new product under an open source license. This model builds upon and extends recent work by Llobet (2003) and Galasso and Schankerman (2010).

Consider a start-up's decision to enter into a software segment by introducing a new OSS product that could generate a revenue  $V$ .<sup>22</sup> Suppose this market segment has a set of patented technologies held by  $n$  different parties that the start-up has to navigate through to successfully enter into the segment. I assume these parties are symmetrical in importance. I examine the case in which the start-up already has access to  $n-1$  of the required patents that would enable it to generate a value  $V' < V$ , and study how the existence of the patent pool affects the  $n$ th negotiation. The start-up's OSS product uses some code developed by the open source community substituting the  $n$ th party's patented technology to make an improvement  $\Delta$  on the code.<sup>23</sup> In other words, product improvements are ordered as a quality ladder, as in O'Donoghue et al. (1998) and Scotchmer (2005). If it enters, the start-up also faces an irreversible investment  $c$  associated with developing and commercializing the new product. Upon the start-up's introduction of this new OSS product, the  $n$ th party can make a settlement offer to the start-up.

My analysis proceeds in two stages. First, I show how the size of the OSS patent pool related to a segment influences the litigation game outcome and consequently the start-up's entry decision. Second, I examine how the marginal effect of the pool is influenced by the value at stake in the  $n$ th negotiation, which is in turn determined by the features of the patent thickets in the market segment. In particular, I focus on two features of patent thickets stressed by prior literature (Noel and Schankerman 2006, Cockburn and MacGarvie 2011). The first is the cumulativeness of innovation within a software segment, defined as the extent to which the innovations within a segment are related with

---

<sup>22</sup> While a firm may produce both open and closed source software (e.g., Bonaccorsi et al. 2006, Lerner and Schankerman 2010), it often does not view these two as substitutes when considering whether to enter a market segment. This viewpoint is consistent with empirical evidence from my data.

<sup>23</sup> Alternatively, the code could be developed internally and issued under an OSS license. While the following discussion still holds under this case, this would add an additional strategic decision for the start-up firm: whether to enter under an open source or proprietary license.

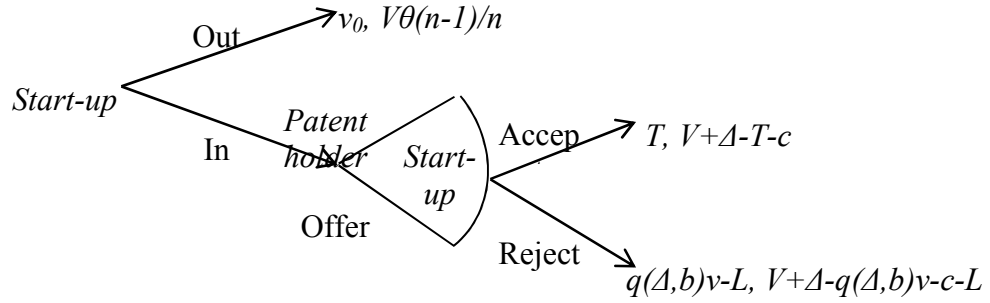
or build upon each other. Second, I study the concentration of patent ownership within a software segment. In my setting, the concentration of patent ownership translates to the number of parties holding the set of essential blocking patents with whom the start-up must negotiate.

Following Galasso and Schankerman (2010) and Lerner and Tirole (2004), I use the following equation to capture the value at stake for the  $n$ th negotiation.

$$v(n, \theta, V) \equiv V - V' = V - V \frac{n-1}{n} \theta \quad (3.1)$$

The value at stake in the  $n$ th negotiation can be written as the difference between  $V$  and  $V'$ . I use  $V' = V \frac{n-1}{n} \theta$  to indicate the value obtainable using only the  $n-1$  patents that are already held by the start-up. Equation (1) shows how characteristics of the patent thicket influence the value of the  $n$ th negotiation. First,  $n$  is the total number of parties that hold blocking patents. A decrease of  $n$  (an increase of concentration of patent ownership) increases the value of the  $n$ th round of negotiation  $v$ , i.e.,  $\partial v / \partial n < 0$ . The parameter  $\theta$  captures the extent of cumulateness in innovation and will influence the value of the  $n$ th negotiation: For example, when innovation in a market segment is very cumulative (lower  $\theta$ ), the value at stake in the  $n$ th negotiation,  $v$ , is very high: it is equal to  $V$  if  $\theta=0$ . In contrast, when innovation in a market segment is non-cumulative (high  $\theta$ ), the patents held by each of the  $n$  parties will independently contribute value to the new product. In the limit, when  $\theta=1$  the value at stake in the  $n$ th negotiation will be equal to  $\frac{1}{n}V$ . Therefore,  $\partial v / \partial \theta < 0$ .

The litigation game is motivated by the model introduced by Llobet (2003). The structure and timing is shown below



In the first stage, the start-up decides whether to enter into a software segment by introducing a new OSS product. If the start-up decides not to enter, given its current access to  $n-1$  patents, its payoff will be  $v = V \frac{n-1}{n} \theta$ <sup>24</sup>; the  $n$ th party will enjoy the amount of  $v_0$  from other potential licensees. If the start-up decides to enter, the patent holder will make a settlement offer to the start-up with a license payment  $T$ . Following Llobet (2003), suppose the size of the improvement  $\Delta$  is private information and unobservable to the patent holder; for simplicity, suppose the patent holder knows that  $\Delta$  is drawn from a uniform distribution  $G(\Delta) = \Delta$  with the density function  $g(\Delta) = 1$ . The patent holder chooses  $T$  to optimize its expected payoff based on the distribution of  $\Delta$ . Therefore, if the start-up accepts the offer, the payoffs will be:

$$\pi_s^s(\Delta, v, T) = V + \Delta - T - c \quad (3.2)$$

$$\pi_p^s(\Delta, v, T) = T \quad (3.3)$$

for the start-up and patent holder, respectively.

If the start-up firm rejects the offer, the patent holder can commit to litigate. Regardless of the outcome of the litigation, both parties incur a litigation cost denoted as  $L_p$  for the patent holder and  $L_s$  for the start-up. For simplicity, I assume  $L_p = L_s = L$ . During the legal proceedings, the courts receive a signal on the true value of the patents and decide whether the start-up's invention is infringing or not based on two factors: (i) how

<sup>24</sup> This value may be realized through a variety of different approaches, such as re-licensing the  $n-1$  parties' patents or attracting venture capital funding.

much the invention improves upon the infringed patent ( $\Delta$ ) and (ii) the validity of the patent holder's patents  $b$ . For simplicity,  $b$  is assumed as  $b \in [0,1]$ :  $b=1$  implies the patents are valid while conversely  $b=0$  implies the patents are invalid. The probability that the courts will favor the patent holder,  $q(\Delta, b)$ , is modeled as the following form for simplicity:  $q(\Delta, b) = b(1-\Delta)$ .<sup>25</sup>

I argue that OSS patent pools can be used to strengthen a start-up's negotiating position when bargaining over blocking patents that are not part of the pool. Because the patents in the pool can be used as prior art during litigation, pool licensees can leverage the pool's patents to invalidate the infringed patents.<sup>26</sup> Such prior art searches have been used to invalidate infringed patents in several well-known cases where OSS was alleged to be infringing, in particular the cases of *Firestar/Datatern v. Red Hat* (Dillon 2008, Paul 2008a) and *Trend Micro v. Barracuda Networks* (Paul 2008b). The greater the number of patents that have been contributed to the pool, the more likely it is that courts will identify prior art from the pool to challenge the nth party's patents.<sup>27</sup> Thus, I expect  $b$  to decrease in the size of  $x$ ,  $\partial b/\partial x < 0$ .

If the start-up is found liable, the court awards the patent holder payment  $v$  from the start-up. This  $v$  represents the payment under the unjust enrichment doctrine, the amount that the start-up would earn from successfully infringing this patent, given that it

---

<sup>25</sup> That is, when  $b=1$  and  $\Delta=0$ , the courts will rule that the start-up is liable and be made to compensate the patent holder. When the start-up makes a significant improvement on the patented technologies, e.g.  $\Delta=1$ , even if  $b=1$ , the courts will rule that the start-up is not liable. On the other hand, when the patent is not valid, i.e.  $b=0$ , no matter how much improvement the start-up makes (for any value of  $\Delta$ ), the start-up will not be liable.

<sup>26</sup> To fulfill this defensive role, the Patent Commons project (the major OSS patent pool within the OSS community) established a partnership with the USPTO to ensure that patent examiners have access to all available prior art in the pool relating to the patents in question. See <http://www.patent-commons.org/news/index.php?displaynews=17&page=1> for more details.

<sup>27</sup> Increases in the size of the patent pool will also strengthen the start-up's negotiating position if it enables the latter to sue the incumbent firm for infringement of one or more of the pool patents. For example, in the *Trend Micro v. Barracuda Networks* case, Barracuda countersued Trend Micro using a portfolio of antivirus patents obtained from IBM (Paul 2008b). As I show below, the pool could also effectively reduce the threshold for OSS entry through countersuits.

had access to the other  $n-1$  proprietary incumbent patents (Galasso and Schankerman 2010, Schankerman and Scotchmer 2001). Thus, the payoff for the patent holder if it wins the lawsuit should be  $v-L$  and for the start-up should be  $V+\Delta-c-L-v$ . If the start-up is found not liable, the patent holder will lose the royalty payment  $v$  for the patent<sup>28</sup> and incur a payoff  $-L$ ; the start-up will receive payoff  $V+\Delta-c-L$ . Therefore, the expected benefits to the patent holder and the start-up if the start-up rejects the settlement offer can be expressed as follows:

$$\pi_s^l(\Delta, v, b) = V + \Delta - q(\Delta, b)v - c - L \quad (3.4)$$

$$\pi_p^l(\Delta, v, b) = q(\Delta, b)v - L \quad (3.5)$$

I denote the strategy of the start-up as  $\sigma_s$ , which includes two decisions: whether to enter by introducing a new OSS product and whether to settle with the patent holder. First, there exists an improvement threshold  $\underline{\Delta}$  that makes the start-up indifferent between not entering (with a payoff  $V\theta \frac{n-1}{n}$ ) and entering. This  $\underline{\Delta}$  can be defined from the equation (3.6) below. Second, there is a threshold of  $\Delta^s(T)$  that makes the start-up indifferent between settlement and going to court: a start-up with improvement larger than  $\Delta^s(T)$  will be more confident that it can win at trial and will therefore choose to go to court; for  $\Delta$  smaller than  $\Delta^s(T)$  but larger than  $\underline{\Delta}$ , the start-up would enter but rather settle. From equations (3.2) and (3.4), I can implicitly define  $\Delta^s(T)$  using equation (3.7).

$$V + \underline{\Delta} - T - c = V\theta \frac{n-1}{n} \quad (3.6)$$

$$T = q(\Delta^s, b)v + L = b(1 - \Delta^s)v + L \quad (3.7)$$

Therefore, the strategy of the start-up as  $\sigma_s$  can be expressed as follows.

$$\sigma \begin{cases} = (\text{out}) \text{ if } \Delta \leq \underline{\Delta} \text{ (i.e., do not enter)} \\ = (\text{in, accept the offer}) \text{ if } \underline{\Delta} \leq \Delta \leq \Delta^s(T) \text{ (i.e., enter and pay amount } T \text{ for licensing fee )} \\ = (\text{in, reject the offer}) \text{ if } \Delta \geq \Delta^s(T) \text{ (i.e., enter and go to trial )} \end{cases}$$

<sup>28</sup> Note that if the patent holder loses, it may not be able to collect the licensing fee  $v$  from any potential licensee. This is because this ruling decision will make it attractive for the potential licensee to adopt the free open source code with functions similar to the patents held by the  $n$ th party but with less cost.

As assumed by Llobet (2003), while the patent holder does not know the size of the improvement  $\Delta$ , it holds a belief  $\tilde{\Delta}$ , above which the start-up will enter. Conditional on this belief, the patent holder will decide  $T$  that maximizes its expected payoff  $\pi(v, b, L | \tilde{\Delta})$

$$\max_T \pi(v, b, L | \tilde{\Delta}) = \max_T \int_{\Delta^s(T)}^1 (q(\Delta, b)v - L) \frac{g(\Delta)}{1-G(\tilde{\Delta})} d\Delta + T \cdot \frac{G(\Delta^s(T)) - G(\tilde{\Delta})}{1-G(\tilde{\Delta})} \quad (3.8)$$

I follow Llobet (2003) to define a pure strategy sequential equilibrium of the litigation game to be a strategy profile  $(T^*, \sigma_s^*)$  such that (i)  $T^*$  maximizes the expected profits for the patent holder in equation (3.8) given a belief  $\tilde{\Delta}$  and (ii) for all  $\Delta$ ,  $\sigma_s^*(\Delta)$  maximizes the expected profit for the start-up given  $T^*$ .

*Lemma 1:* The unique pure strategy sequential equilibrium  $\underline{\Delta}^*$  (the optimal improvement threshold for OSS entry) has the following form<sup>29</sup>:

$$\underline{\Delta}^* = (1 + vb)^{-1}(vb + c - L - v) \quad (3.9)$$

*Proposition 1:* The size of the patent pool  $x$  will decrease the minimum improvement threshold for OSS entry  $\underline{\Delta}^*$ . That is,  $\frac{\partial \underline{\Delta}^*}{\partial x} < 0$ .

The above proposition highlights the effect of an OSS patent pool on OSS entry. By decreasing the likelihood that the courts will uphold the patent,<sup>30</sup> the OSS patent pool will reduce the expected license payment offered by the patent holder. This decreases the threshold of improvement for OSS entry, meaning that start-ups with smaller improvements ( $\Delta$ ) will enter.

---

<sup>29</sup> I provide proofs for this lemma and for all other lemma and propositions in the Appendix B.

<sup>30</sup> If we assume the incumbent's patents to be valid and consider the countersuing effect of the patent pool instead, the start-up will be liable with probability  $1-\Delta$ . However, if the start-up uses the pool patents to countersue, one can interpret the parameter  $b$  as a discount factor for the damage payments  $v$  imposed by the court. That is, the start-up will face the reduced damage payments  $bv$  and the expected payment from the start-up to the patent holder will be  $(1-\Delta)bv$ . For example,  $b$  could represent the effect of ex post negotiation between incumbent and start-up that reduces the size of the damages payments to the incumbent. Mathematically, the model is unchanged and the equilibrium threshold for OSS entry under this alternative interpretation is the same as my baseline model where I model the effects of the patent pool on the infringed patents validity.

I now consider how the efficacy of the patent pool is affected by two features of patent thickets—the cumulativeness of innovation and the concentration of patent ownership. I argue that the key mechanism through which these characteristics influence entry lies in how they shape the incremental value obtained from the focal  $n$ th negotiation (i.e., the size of  $v$ ). Therefore, I first introduce the following lemma related to the interaction of an OSS patent pool and the value at stake for the  $n$ th negotiation:

*Lemma 2:* The threshold for OSS entry  $\underline{\Delta}^*$  will be lower when both the size of the patent pool  $x$  is large and the  $v$  is high. That is,  $\frac{\partial^2 \underline{\Delta}^*}{\partial x \partial v} < 0$ .

Lemma 2 suggests that the marginal effect of the OSS patent pool in lowering the licensing payment  $T$  will be most important when  $v$  is high. Intuitively, consider the start-up's entry decision, which depends on both the value (i.e.,  $\Delta + v$ ) it can obtain from the focal technologies and the potential costs (i.e.,  $T + c$ ). When the size of the patent pool is relatively large, it will reduce the bargaining position of the  $n$ th party, which translates into a lower licensing payment  $T$  for the start-up; further, if the start-up can obtain very high value from the focal technologies, the start-up will become more sensitive to the benefits of the OSS patent pool.<sup>31</sup> These two parameters ( $x$  and  $v$ ) will interact to reduce the improvement threshold for OSS entry.

From equation (1), I know that when the cumulativeness of innovation is high (a low  $\theta$ ), the surplus the start-up can obtain from embedding the focal technologies will be high (a high  $v$ ). Thus, the effect of the OSS patent pool will be stronger when both  $\theta$  is low (a high  $v$ ) and  $x$  is large. Similarly, a higher patent ownership concentration (lower  $n$ ) will also lead to a higher value of the focal technologies (a high  $v$ ) and thus interact with

---

<sup>31</sup> Conversely, any factor that reduces the value of the  $n$ th negotiation will also reduce the effect of the OSS patent pool on entry. For instance, it can be shown that if transaction costs associated with each negotiation (assumed to be zero in the model) are high, the value at stake in the  $n$ th negotiation is lower, and the effect of the patent pool will be reduced. This also suggests that the Proposition 1 is entirely driven by the effect of the OSS patent pool on the bargaining power of the startup.

the effects of the OSS patent pool on the start-up's entry decision. As formally proved in the Appendix B, I have the following propositions.

*Proposition 2:* The threshold for OSS entry  $\underline{\Delta}^*$  will be lower when both the size of the patent pool  $x$  is large and the cumulativeness of innovation is high.

*Proposition 3:* The threshold for OSS entry  $\underline{\Delta}^*$  will be lower when both the size of the patent pool  $x$  is large and the concentration of patent ownership is high

### 3.3 Research Setting

I define an OSS patent pool as a collection of patents pledged to OSS firms for royalty-free usage. I focus on one major OSS patent pool—the OSDL's Patent Commons project (denoted as “the Patent Commons” hereafter). In January 2005, IBM pledged access to its more than 500 software patents to “any individual, community, or company working on or using software that meets the Open Source Initiative (OSI) definition of open source software now or in the future.” Subsequent to IBM's action, several other incumbents that participate in OSS software pledged around 30 patents to this pool.<sup>32</sup> “Pledge” in this context means to offer patents royalty-free to any third party that (i) is engaging in activities that might otherwise give rise to a claim of patent infringement and that (ii) promises not to sue the pool's beneficiaries (Patent Commons project 2005).<sup>33</sup> IBM announced in its press release that it believed this was the largest patent pledge of any kind. The introduction of this pool was expected to confer several benefits upon producers of OSS. Participants in the Patent Commons can freely embed technologies covered by these patents into their own products without any fear of being sued. Further, all pledged patents are explicitly listed on an online public database, and there is no need

---

<sup>32</sup> Example companies include Computer Associates International Inc. and Open Invention Network, LLC.

<sup>33</sup> For more details, see [http://www.patent-commons.org/resources/about\\_commitments.php](http://www.patent-commons.org/resources/about_commitments.php).

to sign any formal agreement between the Patent Commons and the beneficiary of the pool to use them.

It is worth noting that, while I found a series of recent patent pledging events based on a search of major news outlets (see table 3.1 for a detailed summary of these events), my choice of the Patent Commons as the focus of my analysis is guided by several factors. First, the patent pool must be economically important in the sense that it comprises a large collection of patents. Second, since I am interested in measuring how the effect of OSS patent pools vary across different software segments, the patent pool must cover multiple software technology markets. Third, because I expect some time lag between the announcement of the royalty-free patent pool and entry by software startups, the time window between the patent pool contribution and the end of the sampling period should be long enough to observe any significant behavioral changes. Fourth, the patent pool needs to specify the contributed patents on a very detailed level, by, for example, listing the available patent number. The Patent Commons is the only OSS patent pool that met all four of these criteria. Nevertheless, in my empirical analysis I control for the effects of other patent pool-like institutions

**Table 3.1 Patent Pledging Events**

Event Date	Pledging Firm(s)	Pledged Patent(s)	Potential licensees	Notes
Jan 2005	IBM	More than 500 specified patents (contributed to the Patent Commons)	Anyone developing code under an OSI approved license	This pledging event has been included in the analysis
Jan 2005	Sun Microsystems	1670 (unspecified) patents related to Sun's Solaris	Developer working on any approved project under the Common Development and Distribution License (CDDL)	Two main criticisms of this pledge: 1. The CDDL doesn't permit mingling its code with code under GNU GPL, which governs Linux. This means developers can't use these patents on Linux – the freely granted patents can only enable idea-sharing among programmers for Solaris-related projects. 2. Sun's announcement was too broad and didn't specify these 1670 patents or respond to any developers' questions about what rights the developers have to these patents.
Sep 2005	Computer Associates International Inc.	14 patents (contributed to the Patent Commons)	Anyone developing code under an OSI approved license	This pledging event has been included as a control in the analysis.
Nov 2005	Nokia	Any of its patents	Developers working for the Linux Kernel only	Criticism: Because of Nokia's stance on Linux only, developers questioned why it did not apply to directly related projects such as GNOME and KDE and why it did not apply to application projects that are not necessarily directly related to Linux.
Nov 2005	Open Invention Network, founded by IBM, Novell, Koninklijke Philips Electronics, Sony and Red Hat	Any of OIN's patents	Any company, institution or individual that agrees not to assert its patents against the Linux operating system or certain Linux-related applications	This pledging event has been included in the analysis as a control, as many of its patents have been pledged only recently and toward the end of the sample period.
Feb 2007	Blackboard Inc.	Patent 6,988,138, 7,493,396, 7,558,853; pending patent applications: 12/470,739; 10/443,149; 10/643,075; 10/653,074; 11/142,965; 10/373,924; 10/918,016.	Anyone contributing to OSS projects, OSS initiatives, commercially developed open source add-on applications to proprietary products	For the commercially developed open source add-on applications to proprietary products, if the software's end license is open, then it is covered by the pledge; if it is partly open and partly proprietary, it is not covered.

## 3.4 Data

### 3.4.1 Sample

The sample consists of 2,054 US software firms from the 2004 and 2010 editions of the CorpTech Directory of Technology Firms<sup>34</sup> (denoted as CorpTech 2004/2010 hereafter) that primarily operate in the prepackaged software industry. I combine this sample with data from the National Establishment Time Series (NETS) Database, which includes 100,000 US-based firms with primary SIC 7372. The use of two data sources reflects constraints with each. The CorpTech data have detailed information on the product market segments of firms, but have little time variation, while the NETS data have limited product market information but vary over time.

As noted above, the focus of my study is on start-up firms. As a result, I restrict the sample to firms that were founded after 1990 and that have fewer than 1000 employees and less than \$500 million in annual sales.<sup>35</sup> The sampling period is from 1999 to 2009, with 6 years before the establishment of the Patent Commons and 5 years after. I believe this time window is sufficiently long to capture the impact of the Patent Commons on OSS entry.

### 3.4.2 Identifying Software Segments And The Matching Patent Classes

I use the product code classification system embedded in the Gale database “PROMT” (Fosfuri et al. 2008) as my primary source to define software market

---

<sup>34</sup> My choice of 2010 CorpTech data reflects a constraint with the data—I have contacted CorpTech and there are no historical data between 2005-2009, the core years of my sample period. The combination of CorpTech 2004 and 2010 is to address potential survivor bias.

<sup>35</sup> The results are robust to the use of alternative thresholds for inclusion in my sample. For example, the results are robust to an alternative sample of start-ups that includes firms founded after 1990 that have fewer than 500 employees and less than \$500 million annual sales.

segments.<sup>36</sup> Because of two major drawbacks of relying only on the PROMT classifications (I describe these in further detail in the Appendix C), I further consolidate PROMT product categories with CorpTech’s “SOF” product classes to create a PROMT-CorpTech concordance. Under this concordance, each PROMT product code is associated with a detailed set of keywords. The keywords for each product class are used to (i) manually assign PROMT product codes to PROMT news articles with missing codes and (ii) match software segments with the most relevant patent classes as described below.

An important part of my data construction involves matching product market segments to patents. This allows me to identify both the cumulateness of innovation and the concentration of patent ownership in a software segment. As is well-established in the literature, this matching is a challenging undertaking (e.g., Griliches, 1990, Silverman 1999). To facilitate the mapping between software products and patents, I follow Cockburn and MacGarvie (2006, 2011) and match software patents to CorpTech “SOF” product classes<sup>37</sup> to create a patent-CorpTech concordance. Because the software segments are classified through PROMT categories, in order to create the final mapping between software segments and patent classes, I then combine the PROMT-CorpTech concordance and patent-CorpTech concordance to form the PROMT-patent concordance. The final concordance that I use in the empirical analysis consists of 33 software segments matched to 422 patent class-subclass combinations<sup>38</sup> (see the Appendix C for a detailed discussion of my data construction process).

---

<sup>36</sup> A few examples of PROMPT product codes are included in Table C.1 in Appendix C.

<sup>37</sup> There are more than 290 software product codes (denoted as SOF) defined by CorpTech Directory. Each firm in this directory is associated with a set of self-reported product codes selected from these 290 SOF categories.

<sup>38</sup> Table C.3 in Appendix C lists examples of this final concordance between software segments and US patent class-subclass combinations.

## 3.5 Measures

### 3.5.1 Dependent Variable: OSS entry

This variable refers to the number of OSS entrants into software segment  $j$  in year  $t$ . It is to capture the threshold to entry ( $\underline{\Delta}^*$ ) in my propositions. I use a three-step procedure to identify new OSS entry in a software segment based on the press releases of the 2,054 firms in the PROMT database. First, following work by Fosfuri et al. (2008) and Bessen and Hunt (2007), I searched for a set of keywords within PROMT articles to identify articles related to OSS. Appendix D includes the full set of keywords. Second, I *manually* read all search results that included words from the first step to identify new OSS product introductions. I identified an article as referring to an introduction of a new OSS product when the article indicated that either of the following took place: (i) the introduction of a new software product that offered one or more of its module(s)<sup>39</sup> under an open source license (I label such modules as *open source modules*); and (ii) the introduction of a new version of an existing software product with open source modules. Third, to identify entry I kept only the first open source module release by a start-up into a segment. In total, I have 242 new OSS product entries made by 85 start-up firms from 1999 to 2009.<sup>40</sup> I aggregated these new OSS product entries by software segment and year. The dependent variable is therefore equal to the number of new OSS start-up entrants in segment  $j$  and year  $t$ . The data are structured as a balanced panel. Table 3.2 includes a brief description of measures and summary statistics for all variables used in my empirical analysis.

---

<sup>39</sup> In software, a module is a part of a program. A software product is composed of one or more modules that are linked together but perform different functions (e.g. the calendar module available in the Microsoft Office's Outlook).

<sup>40</sup> This procedure implicitly assumes there is no OSS entry by firms prior to 1999. I believe this assumption is supported by empirical evidence. For example, SourceForge, a major repository of OSS, was started in November 1999.

### 3.5.2 Independent Variables

*OSS patent pool.* This variable refers to the number of patents that belong to the Patent Commons that are related to software segment  $j$  in year  $t$ . It corresponds to the OSS patent pool size  $x$  in the model. As discussed by Merges and Nelson (1990), it is the scope of a patent that determines the patent's economic and legal significance. In a setting with cumulative technologies, broader pool patents will be more likely to invalidate blocking patents. To capture these effects, I measure the claims-weighted count of patents in the Patent Commons pool related to each software segment.<sup>41</sup> I further take the logged value of this variable<sup>42</sup> to reduce skewness.

*Cumulativeness.* This variable refers to the cumulativeness of innovation within segment  $j$  in year  $t$  and is negatively correlated with the parameter  $\theta$  in the model. To measure this concept I use patents' backward citations, which provide information about "existing ideas used in the creation of new ideas" (Caballero and Jaffe 1993) and indicate "some form of cumulative technological impact" (Jaffe et al. 1998). Following Clarkson (2005), I measure it based on the average propensity for patents in segment  $j$  and year  $t$  to backward cite patents within the same segment  $j$ . This is roughly similar to the way economists have measured the cumulative nature of innovation at the firm level, e.g. using the extent to which firms self-cite their own patents (Hall et al. 2005). In my setting, I proceed as follows. If I sort the  $N$  patents within a software segment  $j$  chronologically (with  $m=1$  being the oldest patent and  $m=N$  being the youngest), the cumulativeness for each patent  $n$  (i.e. the propensity for patent  $n$  to cite preceding patents within the same segment) is calculated as  $C_n = \sum_{m=1}^N \frac{x_{nm}}{n-1}$ , where  $x_{nm}$  is a dummy variable equal to one if patent  $n$  back-cites patent  $m$ , and zero otherwise (with both

---

<sup>41</sup> I also use raw patent counts as a robustness check. The results are qualitatively similar to this claims-weighted measure.

<sup>42</sup> I add 1 to the variable when taking the log.

patents belonging to the same segment),  $(n-1)$  is the total number of possible citations, and  $n > 1$ , since  $C_1$  is undefined. In other words, the cumulateness of a focal patent in segment  $j$  is based on the share of potential backward citations to patents belonging to the same segment that are actually cited by the focal patent. The cumulateness of innovation for software segment  $j$  is then the average of all  $N-1$  patents' cumulateness:<sup>43</sup>  $C_j = \frac{\sum_{n=2}^N \sum_{m=1}^N \frac{x_{nm}}{n-1}}{N-1}$ . This measure varies over time based on the grant year of the segment  $j$  patents under consideration. Notice that the oldest patents in a segment tend to have greater cumulateness since the potential number of patents that can be cited is smaller. As a robustness check, I also used an alternative weighting scheme, one that provides relatively lower importance to the cumulateness measure of older patents. As in Clarkson (2005), it is calculated as  $C_j = \frac{\sum_{n=1}^N \sum_{m=1}^N x_{nm}}{N(N-1)/2}$ . For both measures, I take the logged value to reduce skewness.

*Concentration.* This variable indicates the extent of concentration of patent ownership in a segment and is negatively correlated with the parameter  $n$  in the model. Following Noel and Schankerman (2006) and Cockburn and MacGarvie (2011), I use the four-assignee citation concentration ratio to measure the concentration of patent ownership in a software segment. Backward citations indicate the extent to which a technological area has already been covered by prior art, so the share of backward citations owned by an assignee suggests the extent to which the assignee holds existing patented technologies and therefore the importance of negotiating with the assignee. To construct this variable, I first calculate the number of citations made by patents in segment  $j$  up to year  $t$  that are held by the cited assignee  $n$  (denoted as  $s_{njt}$ ). Then I arranged  $s_{njt}$  in descending order. The total citations owned by the four firms that

---

<sup>43</sup> The average only considers  $N-1$  patents since  $C_1$  is undefined.

received the top four largest number of citations made by patents in segment  $j$  in year  $t$  (i.e. the top four  $s_{njt}$ , where  $n=1,2,3,4$ ) is  $\sum_{n=1}^4 s_{njt}$ . Thus, the four-assignee citation concentration ratio for segment  $j$  in year  $t$  is calculated as  $\frac{\sum_{n=1}^4 s_{njt}}{total\_citations_{jt}}$ , where  $total\_citations_{jt}$  is the total number of citations made by patents in segment  $j$  up to year  $t$ .<sup>44</sup>

### 3.5.3 Control Variables

*Sales.* One important factor that may correlate with both the behavior of firms contributing to the Patent Commons and OSS entry by start-ups is the size of the market in software segment  $j$ , which is proxied by the total sales in segment  $j$  in year  $t$ . Because I do not have CorpTech data between 2005 and 2009, I use NETS data to measure this variable. Roughly 4,500 software firms in the NETS data are assigned to one of the eight-digit SIC categories (e.g., 73729901) that correspond to eight broad categories in the software industry. I compute the total sales for each of the eight SIC categories and then map them to the 33 software segments and use the matched sales to approximate the overall sales for each segment.

*Potential licensors.* The costs associated with patent thickets are determined not only by the concentration of patent ownership in a segment, but also by the total number of different parties holding patents essential to a segment. I compute the total number of assignees who hold citations made by all patents in segment  $j$  up to year  $t$  to measure the total potential licensors with whom a start-up would have to negotiate. I further use the logged value to reduce the skewness of this variable.

---

<sup>44</sup> I also use an eight-assignee citation concentration ratio as a robustness check. The results are qualitatively similar to this four-assignee citation concentration ratio measure.

*Total patents.* Although I am most interested in two of the most important features of patent thickets—the cumulativeness of innovation and the concentration of patent ownership, the total number of patents related to a market has also been used as a measure of the density of patent thickets (Cockburn and MacGarvie 2011). I add this variable as an additional control to isolate this effect, computing the claims-weighted patent count related to each software segment  $j$  cumulated up to year  $t$ .

*Patent quality.* This variable is a control for the quality of patents in the market segment  $j$  in year  $t$ . As has been noted elsewhere, higher quality patents suggest superior technological capabilities possessed by existing incumbents in the segment, which leaves less room for start-ups to innovate further. This variable is equal to the log value of the cumulative stock of citations received by the patents in segment  $j$  (adjusted for truncations) divided by total number patents in  $j$  up to year  $t$ .

*Open Innovation Network (OIN) patents.* At the end of the sample period, another OSS patent pool institution—OIN—was established. Similar to the Patent Commons, OIN offers contractually royalty-free usage of these patents to OSS participants as long as users promise not to file suit against software associated with the Linux System.<sup>45</sup> I do not focus on this pool in my main analysis as it was introduced too late in my sample period to have a measureable effect on entry during the sample.<sup>46</sup> However, I include it as a control. I measure this variable as the claims-weighted patent count of OIN patents related to software segment  $j$  cumulated up to year  $t$ .

*Standard-setting organization (SSO) patents.* As mentioned earlier, another important mechanism to address the anti-commons problem is SSOs. Such institutions promote coordination of innovation by providing a forum for collective decision-making

---

<sup>45</sup> For the detailed definition of the Linux system, see [http://www.openinventionnetwork.com/pat\\_linuxdef.php](http://www.openinventionnetwork.com/pat_linuxdef.php).

<sup>46</sup> For the 130 patents contributed to OIN from year 2006 to 2009, 70 percent were contributed in 2008 and 2009.

among firms, facilitating the introduction of standards (Rysman and Simcoe (2008)). If any patent is incorporated into the standards, the patent owner can gain significant power to control the diffusion of such standards and even deter market entry (Shapiro 2001, Rysman and Simcoe 2008). To prevent this blocking effect, most SSOs require patent holders contributing to the standard to license their patents on “Fair, Reasonable, and Non-Discriminatory (FRAND)” terms. Firms can even choose to license their patents with FRAND and royalty-free terms. I control for the incidence of SSO patents that are licensed royalty-free because I expect that such patents might also have some effect on OSS entry. Therefore, I collect all patents disclosed under royalty-free licenses by the major eight SSOs (e.g., IEEE, ITU) from 1971 to 2008 (Rysman and Simcoe 2008)<sup>47</sup>. I compute the claims-weighted patent count of the SSO patents that are distributed under royalty-free licenses and are related to software segment  $j$  cumulated up to year  $t$ .

---

<sup>47</sup> I am grateful to the generous offer of the SSO patent data set by Tim Simcoe and Christian Catalini. These data are available for download under a creative commons license at [www.ssopatents.org](http://www.ssopatents.org).

**Table 3.2 Summary Statistics, Chapter 3**

Variable name	Measure (Segment-year)	Obs.	Mean	Std. Dev.	Min	Max
OSS entry	The number of new OSS entrants into segment j in year t	363	.667	1.344	0	11
OSS patent pool	Log of Patent Commons' claims-weighted patent count related to segment j cumulated up to year t	363	2.413	2.936	0	7.911
Cumulativeness	Log of cumulativeness of innovation in segment j up to year t-1	363	.808	.613	.095	3.454
Cumulativeness_R	Log of cumulativeness of innovation in segment j up to year t-1 using the robustness measure proposed by Clarkson (2005)	363	.473	.490	.036	2.933
Concentration	Four-assignee citation concentration ratio in segment j up to year t-1	363	0.227	0.075	0.076	0.458
Potential licensors	Log of total number of assignees (divided by 100) that are cited by patents in segment j up to year t-1	363	5.559	1.301	1.072	8.407
OIN patents	Log of Open Invention Network's claims-weighted patent count in segment j cumulated up to year t	363	1.125	2.053	0	6.690
SSO patents	Log of standard-setting organizations' claims-weighted patent count in segment j cumulated up to year t	363	1.628	2.118	0	5.908
Total patents	Log of total claims-weighted patent count related to segment j cumulated up to year t-1	363	10.817	1.232	6.870	13.486
Patent quality	Log of average quality of patents related to segment j cumulated up to year t-1	363	2.832	.402	1.839	4.051
Sales	Log of total volume of sales by segment j in year t (in Million)	363	7.223	1.523	2.660	9.145
IBM patents * year2003_2004	Log of IBM's pre-sampling stock of claims-weighted patent count * year2003_2004 dummy, where year2003_2004 dummy turns on for year t = 2003, 2004	363	1.209	2.708	0	9.838
IBM patents * after_year2005	Log of IBM's pre-sampling stock of claims-weighted patent count * after_year2005 dummy, where after_year2005 dummy turns on for year t = 2005, 2006, ..., 2009	363	3.023	3.584	0	9.838
IBM patents opposed at EPO	Log of IBM's patents granted by the European Patent Office (EPO) and opposed at EPO related to segment j cumulated up to year t	363	1.596	.829	0	3.367
IBM patents * year 2003	Log of IBM's pre-sampling stock of claims-weighted patent count * year2003 dummy, where year2003 dummy turns on for year t = 2003	363	.605	2.009	0	9.838
IBM patents * after_year2004	Log of IBM's pre-sampling stock of claims-weighted patent count * after_year2004 dummy, where after_year2004 dummy turns on for year t = 2004, 2005, ..., 2009	363	3.627	3.635	0	9.838

### **3.6 Empirical Strategies And Results**

I motivate the empirical analyses by first investigating the value of patents in the OSS patent pool. If the patents in the pool are not valuable, then they will have little influence on start-up behavior. Next, I establish a relationship between the changes in OSS entry by start-up firms and the changes in the size of the OSS patent pool. I initially assume that any changes in the number of pool patents are uncorrelated with unobservables influencing new OSS entry; I then relax this assumption through instrumental variables estimation. Finally I show that the marginal impact of the OSS patent pool is greater in segments where the cumulativeness of innovation is high or the concentration of patent ownership is high.

#### **3.6.1 Are OSS Patent Pools Less Valuable Than Comparable Patents?**

In my first set of analyses, I examine the quality of patents in the pool relative to a comparison group. Following the matching method employed by Jaffe et al. (1993) and used by many others, I construct a comparison group of patents by choosing the non-pool patents that belong to the same three-digit patent US class as each pool patent and were granted either in the 2 years before the grant year or in the 2 years after the grant year of each pool patent. Table 3.3 presents how patents in the pools compare to comparable patents with respect to forward citations, backward citations, and number of claims. As shown in the first row of table 3.3, pool patents' forward citations are significantly higher than those of control patents. However, non-pool control patents have a significantly higher number of backward citations and claims. This comparison may suggest that while pool patents may cover a narrower technology scope than similar non-pool patents, the

pool patents are indeed valuable in the sense that they are less derivative than other comparable patents and are cited more frequently.<sup>48</sup>

**Table 3.3 Pool Patents Compared to Non-pool Patents**

		Pool Patents	Non-pool Control Patents	T-test
	Obs	2117	250618	
Forward citations	Mean (Std.Err.)	24.927 (0.818)	12.089 (0.040)	15.680***
Backward citations	Mean (Std.Err.)	11.121 (0.364)	16.081 (0.052)	-13.504***
Claims	Mean (Std.Err.)	20.019 (0.310)	20.897 (0.028)	-2.821***

Note: 1) Forward citations are the forward citations as of Dec 31, 2009 and are adjusted for truncation based on the methods by Hall et al. (2001); 2) Following the matching method employed by Jaffe et al. (1993) and followed by many others, I construct the sample of non-pool control patents by choosing the non-pool patents that belong to the same three-digit class as each of the pool patents and were granted either in the 2 years before the grant year or in the 2 years after the grant year of each pool patent; 3) \*\*\*: significant at 1%.

---

<sup>48</sup> I acknowledge that just like the patents disclosed in SSOs (Rysman and Simcoe 2008), one other potential reason that pool patents' forward citations are larger than those of comparable patents is the effect of the patent pool on improving the awareness of these patents, which makes either OSS or non-OSS participants more likely to cite these patents. Thus, the forward citations could be more a measure of follow-up innovations building upon the pool patents rather than a measure of patent quality

### 3.6.2. Does The OSS Patent Pool Encourage OSS Entry?

Proposition 1 shows how changes in the size of an OSS patent pool influence the threshold for entry. The testable implication in my data is how changes in the size of the pool influence the number of entrants. My empirical approach is motivated by recent research that has studied how patent thickets influence market entry in the software industry (e.g., Cockburn and MacGarvie 2011). There are several identification challenges in interpreting a relationship between OSS pool patents and OSS entry as a causal one. First, the entry rate and the stock of pool patents may be co-determined by some unobserved segment-specific factor such as variance in demand across different market segments or the stage of the industry life cycle. To mitigate this concern, my focus is on the time series variation in the size of patent pools within a software segment and its interaction with the segment-specific patent thicket variables. That is, I model new product entry using count data models with conditional fixed effects. Suppose the number of OSS entrants in software segment  $j$  in year  $t$  (denoted as  $Y_{jt}$ ) follows a Poisson process with parameter  $\lambda_{jt}$  taking the form  $\lambda_{jt} = \exp(X_{jt}'\beta)$ . Also suppose  $\alpha_j$  is a segment-specific and time-constant variable that incorporates unobserved heterogeneity across segments. Thus,  $E(Y_{jt} | X_{jt}, \alpha_j) = \lambda_{jt} = \alpha_j \exp(X_{jt}'\beta)$ , and I assume

$$X_{jt}'\beta = \beta_1 OSS\ patent\ pool_{jt} + \gamma_1 Sales_{jt} + \gamma_2 SegmentPatents_{jt-1} + \gamma_3 PatentThicket_{jt-1} + \gamma_4 OtherFreePatents_{jt} + \tau_t \quad (2.10)$$

The vector  $SegmentPatents_{jt-1}$  includes the *Total patents* <sub>$jt-1$</sub> , *Patent quality* <sub>$jt-1$</sub> , and *Potential licensors* <sub>$jt-1$</sub> ; the vector  $PatentThicket_{jt-1}$  includes the two patent thicket variables *cumulativeness* <sub>$jt-1$</sub>  and *concentration* <sub>$jt-1$</sub> . The two vectors are lagged by one year to allow for any lagged effects on OSS entry. The vector  $OtherFreePatents_{jt}$  represents the patents from OIN and SSOs—*OIN patents* <sub>$jt$</sub>  and *SSO patents* <sub>$jt$</sub> .  $\tau_t$  includes 10 year dummies to control for time-varying factors that may influence OSS entry. The model is then estimated using maximum likelihood with robust standard errors clustered at the segment level. I am interested in the estimate for  $\beta_1$  which, if positive, supports

proposition 1. To test the robustness of my results, I use different specifications by adding the above controls incrementally.

Table 3.4 presents the estimation results for specification (2.10). Column (1) shows the simplest specification with only segment sales as a control as well as with segment and year fixed effects. I include sales in all specifications because both demand and market competition within a segment are important determinants of start-up entry.<sup>49</sup> The coefficient in column (1) suggests that a 10% increase in the OSS patent pool's patent claims related to a software segment is associated with a 1.4% increase in OSS entry in that segment. Results are robust when I add controls such as the segment's patent size and quality (the vector  $SegmentPatents_{jt-1}$ ), the segment's patent thicket density (the vector  $PatentThicket_{jt-1}$ ), and the size of the patents included in other patent pool-like institutions (the vector  $OtherFreePatents_{jt}$ ). I note that while increases in the size of the OSS patent pool are associated with OSS entry, increases in SSO patents are not. I speculate that this may reflect differences in the licensing requirements for patent pool and SSO patents: in particular, while users of the patent pool pledge not to sue the pool's beneficiaries, licensees of SSO patents have no such requirements. Licensees of SSO patents may see the value of complementary IPRs increase in value, which may increase their incentives to defend their technologies more aggressively. Thus, increases in SSO patents may not reduce the costs of OSS entry.

---

<sup>49</sup> I have experimented with other controls for market demand such as the number of incumbents. Regressions using these other controls yield qualitatively similar results for the main parameters of interest.

**Table 3.4 Conditional Fixed-effect Poisson**

Dependent variable: OSS entry			
	(1)	(2)	(3)
OSS patent pool	.141* (.081)	.160** (.087)	.154* (.091)
Sales	-.003 (.611)	-.045 (.586)	-.045 (.536)
Potential licensors		.715 (1.697)	1.217 (1.654)
Total patents		-1.770 (2.427)	-1.895 (2.527)
Patent quality		-1.140 (2.391)	-1.242 (2.391)
OIN patents			-.057 (.076)
SSO patents			-.015 (.074)
Cumulativeness			1.152 (1.134)
Concentration			-9.348 (6.818)
Observations	286	286	286
Log pseudolikelihood	-229.734	-229.201	-227.777

Notes: 1) Robust standard errors, clustered by market segment, are in parentheses. 2) \* significant at 10%, \*\* significant at 5%, \*\*\* significant at 1%. 3) The number of observations is lower than 363 (i.e., the number of observations listed in Table 1) because of the use of conditional fixed effects Poisson models, which drops market segments without OSS entry over the entire sample period. 4) All regressions include market and year fixed effects.

One potential concern with the above specification is that there may exist unobserved changes specific to the software segment that are correlated both with contributions to the pool and OSS entry. One such possibility is if product market growth is inadequately controlled for by the *Sales* variable. To address this concern, I use a moment-based count data model with instrumental variables and solve the moment conditions through Generalized Method of Moments (GMM) estimation. That is, since the conditional mean  $E(Y_{jt} | X_{jt}, \alpha_j)$  is equal to  $\alpha_j \exp(X_{jt}'\beta)$ , it implicitly defines the following regression model (Windmeijer and Santos Silva 1997):

$$\begin{aligned} Y_{jt} &= \alpha_j \exp(X_{jt}'\beta) + u_{jt} = \alpha_j \exp(\beta_1 OSS\ patent\ pool_{jt} + \gamma_1 Sales_{jt} + \\ &\gamma_2 SegmentPatents_{jt-1} + \gamma_3 PatentThicket_{jt-1} + \gamma_4 OtherFreePatents_{jt} + \tau_t) + u_{jt} \\ &= \mu_{jt} \alpha_j + u_{jt} \end{aligned} \quad (3.12)$$

where  $\mu_{jt} = \exp(\beta_1 OSS\ patent\ pool_{jt} + \gamma_1 Sales_{jt} + \gamma_2 SegmentPatents_{jt-1} + \gamma_3 PatentThicket_{jt-1} + \gamma_4 OtherFreePatents_{jt} + \tau_t)$  and  $u_{jt}$  is the error term. *OSS patent pool<sub>jt</sub>* is treated as a potentially endogenous variable. Suppose  $X_{jt}'\gamma = \gamma_1 Sales_{jt} + \gamma_2 SegmentPatents_{jt-1} + \gamma_3 PatentThicket_{jt-1} + \gamma_4 OtherFreePatents_{jt} + \tau_t$ , then  $X_{jt}$  is assumed to be exogenous. Following Windmeijer (2000) and Kim and Marschke (2005), I use Wooldridge's quasi-differencing transformation (Wooldridge 1997) to remove the segment-specific fixed effect, and obtain the following moment condition:

$$E\left(Z_{jt} \begin{pmatrix} \frac{Y_{jt}}{\mu_{jt}} - \frac{Y_{jt-1}}{\mu_{jt-1}} \end{pmatrix}\right) = 0 \quad (3.12)$$

where  $Z_{jt}$  includes the set of exogenous variable  $X_{jt}$  and a set of instruments for *OSS patent pool<sub>jt</sub>* as detailed below. As noted by Wooldridge (1997), one drawback for this moment condition is that the estimates of the associated coefficients tend to go infinity if the explanatory variables contain only nonnegative values, as is the case for my data. One solution to this problem proposed by Windmeijer (2000) is to transform  $Z_{jt}$  as deviations from the overall sample mean; therefore, I transform all  $Z_{jt}$  to  $Z_{jt} - \bar{z}$ , where

$$\bar{z} = \frac{1}{NT} \sum_{j=1}^N \sum_{t=1}^T Z_{jt}.$$

The first instrumental variable (IV) is the pre-sampling stock of patents at the end of 1997 held by the major contributor to the Patent Commons—IBM—across the 33 software segments. This instrument is designed to capture IBM’s propensity to pledge patents across different segments. I expect that incumbents tend to contribute patents to segments where they have accumulated stocks of patents for blocking rivals and facilitating negotiations. A shortcoming with this variable is that its variance is cross-sectional, which would result in its being eliminated by the quasi-differencing method. Thus, I further interact the pre-sampling stock of patents with two time dummies associated with the formation of the OSS patent pool. The first time dummy (denoted as *year2003\_2004*) is turned on for year 2003 and year 2004. The motivation to use this time dummy is driven by the observation that on March 7, 2003 IBM was sued by the SCO Group, which asserted that the Linux system embedded by IBM infringed on SCO’s UNIX System V source code. This was the first major IPR enforcement lawsuit targeting OSS that attracted significant publicity and as such, it is expected to influence IBM’s patent contribution decision. The second time dummy (denoted as *afteryear2005*) is set to be equal to 1 after 2005. This time dummy is designed to reflect the concentration of patent-pledging events for the OSS community during 2005. As described in the Appendix B and also shown by Alexy and Reitzig (2011), following the publication of a report by Open Source Risk Management (OSRM) that Linux had been infringing 283 patents, a series of non-assertion announcements by software industry incumbents came out beginning in 2005. Further, as suggested by Alexy and Reitzig (2011), these incumbents had strong incentives to coordinate with each other to avoid the hold-up problems for the OSS community, resulting in a cluster of non-assertion announcements. Thus, I believe that interacting the pre-sampling stock of IBM patents with the timing of these two events will capture any variance in the propensity of IBM to contribute patents to this pool.

The second instrumental variable is the cumulated number of IBM patents that were opposed at the European Patent Office (EPO) up to year  $t$  and related to software segment  $j$  (denoted as *IBM patents opposed at EPO*). The logic to this instrument is that firms will contribute patents that have the potential to block OSS innovation. Therefore, measures that are correlated with the propensity for IBM to create blocking patents related to a software product market segment over time are, therefore, potential instruments. I argue that one proxy for this propensity is IBM's patents that are opposed at the European Patent Office (EPO). Different from US legal procedures, the EPO allows patents granted at the EPO to be opposed up to nine months after the grant date and at a much lower cost than that of patents opposed through formal legal procedures. Using this instrument has two advantages. First, since the opposition is filed at the EPO, it should not be correlated with new OSS product entries in the US market. Second, since the opposition needs to be filed within 9 months from the grant of a patent, the truncation issue associated with the lag between the grant date of a patent and the timing of its impact on innovation is far less serious than it is with other similar procedures such as litigation events.<sup>50</sup>

Table 3.5 presents the GMM estimation results where I instrument for *OSS patent pool*<sub>jt</sub> using the above instruments. To improve identification, I also add the square of the instruments described above (Gallant 1987). I also test for the validity of all instruments used. As before, I use different specifications by adding the controls incrementally. That is, column (1) in table 3.5 provides the estimates using only sales as controls; column (2) further adds the vector *SegmentPatent*; column (3) further adds the vector

---

<sup>50</sup> More specifically, as has been noted by Hall et al. (2003), because oppositions must be filed within 9 months of a patent grant, the average lag between applying for a patent and the filing of opposition request is relatively tight; in contrast, other legal procedures such as litigation and patent re-examinations can be initiated at any time during the lifetime of a patent, which results in greater variance in the distribution of these procedural lags.

*PatentThicket*<sup>51</sup>. As I can see from column (1) to column (3), the estimated direct impact of the OSS patent pool is consistently and significantly positive across all specifications, though inclusion of *SegmentPatent* and *PatentThicket* as controls reduces the magnitude of the effect. Note that since the use of the conditional fixed effects model excludes market segments without OSS entry over the sample period, I also present comparable GMM estimates obtained by dropping these segments. The results are presented in column (4) to column (6) of Table 3.5, and are consistent and similar to the estimates based on the full sample.

---

<sup>51</sup> I was not able to add the vector *OtherFreePatents*, as it leads to non-convergent results.

**Table 3.5 GMM Estimates with IVs**

Dependent variable: OSS entry	Full sample			Sample dropping the segments without OSS entry		
	(1)	(2)	(3)	(4)	(5)	(6)
OSS patent pool	.430*** (.139)	.241* (.127)	.249** (.102)	.406*** (.153)	.258** (.132)	.212* (.111)
Sales	.100 (.345)	-.178 (.153)	-.397* (.215)	.132 (.348)	-.128 (.149)	-.298 (.217)
Potential licensors		2.094 (2.092)	3.260 (2.045)		1.907 (2.142)	2.782 (2.270)
Total patents		-1.750 (2.186)	-2.773 (1.872)		-1.537 (2.220)	-2.455 (2.037)
Patent quality		1.212 (.843)	2.764*** (.869)		1.464* (.757)	2.853*** (.903)
Cumulativeness			-.404 (.924)			-.700 (1.119)
Concentration			-8.671 (6.961)			-7.488 (7.213)
Year dummies	Yes	Yes	Yes	Yes	Yes	Yes
Observations	363	363	363	286	286	286
Over-identification test statistic (p-value)	10.303 (0.24)	15.313 (0.17)	17.442 (0.10)	10.521 (0.23)	15.030 (0.18)	17.097 (0.11)

Notes: 1) The full set of IVs include IBM patents\*year2003\_2004, IBM patents\*afteryear2005, square of IBM patents\*year2003\_2004, square of IBM patents\*afteryear2005, IBM opposed patents at EPO, and square of IBM opposed patents at EPO. 2) Robust standard errors, clustered by market segment, are in parentheses. 3) \* significant at 10%, \*\* significant at 5%, \*\*\* significant at 1%. 4) Year dummies include year2001\_2002, year2003, year2004, year2005, year2006, year2007, and year2008\_2009. The reason for not using the full set of ten year dummies is that including more year dummies leads to non-convergence of the GMM estimator.

I probe the validity of the instruments by first examining the power of the instruments, by using the F-test from an auxiliary first stage ordinary least square (OLS) regression of the endogenous variable against the same set of IVs and exogenous variables. As shown in table 3.6, the F-statistics on the coefficients of my instruments range from 35.17 to 83.63, depending upon the specification, and are all statistically significant, which suggests that the instruments have some power in explaining the endogenous variable. I also perform the Hansen J statistic to test the over-identification restrictions. The results are in the last row of table 3.5. All tests fail to reject the null that the instruments used are uncorrelated with the error term across all specifications. I also test the validity of a subset of the instruments. In particular, I assume that the stock of IBM patents opposed at EPO is exogenous, and test the validity of the interactions between the pre-sampling stock of IBM patents with the year dummies.<sup>52</sup> As shown in table 3.6, based on the C-test statistic I fail to reject the null that the IVs—the *pre-sampling stock of IBM patents* times *year2003\_2004* and its square and the *pre-sampling stock of IBM patents* times *afteryear2005* and its square—are valid across all specifications.<sup>53</sup>

As an additional robustness check, another set of IVs has been constructed: I interact the pre-sampling stock of IBM patents with a year 2003 dummy (which is set to be 1 for year 2003 and 0 for other years) and an afteryear2004 dummy (which is set to be 1 after year 2004 and 0 for other years); I use these two plus their squares and IBM

---

<sup>52</sup> I implement a C-test to evaluate the exogeneity of this subset of IVs, with the null hypothesis that they are valid instruments. That is, I treat the full model with the three IVs and their squares as the restricted and fully efficient regression; I use the model with the IBM patents opposed at EPO and its square as IVs as the unrestricted, inefficient but consistent regression.

<sup>53</sup> Alternatively, I also performed a test on the validity of the subset of instruments represented by the stock of IBM patents opposed at the EPO, and this time assume that the interactions between the pre-sampling stock of IBM patents and the year dummies are exogenous. Based on the C-test statistic, I fail to reject the null that the stock of IBM patents opposed at the EPO is valid across all specifications.

opposed patents at EPO plus its square as IVs for GMM estimation using the same model. The results are similar and presented in table 3.7.

Finally, to further boost confidence in my results, I also implement a falsification exercise. The intuition is that, given the theoretical predictions, I should not observe a positive effect of the OSS pool on start-up entry based on proprietary software product entry. I use the press releases of the 2,054 firms in the PROMT database to identify new proprietary software product entry. To precisely identify new proprietary products related to each segment, I focus only on introduction events associated with PROMT product codes. For each start-up, I include only the firm's first product in a segment to capture entry. This results in 2,384 proprietary product entry events from 2002 to 2009. I then aggregated these by software segment and year. I also adjust for a change in the assignment of product codes during the sample. Specifically, between 2007 and 2009 I found that application-related software products were systematically assigned to a higher product code level in the PROMT database (specifically, they were assigned to 7372400, Applications Software). This forced me to combine several application segments together, leaving me with 29 software segments (rather than the baseline 33 segments) in total. As shown in table 3.8, the results suggest that there is no significantly positive effect of the pool on entry using this set of products.

**Table 3.6 C-test, First Stage Results of OLS IV Regressions**

	(1)	(2)	(3)
<i>GMM Estimates</i>			
C-test statistic (p-value)	10.032 (0.123)	3.047 (0.803)	5.015 (0.542)
<i>First Stage Results of OLS IV Regressions</i>			
Dependent variable: OSS patent pool			
IBM patents*year2003_2004	.010 (.099)	.090 (.100)	.090 (.095)
square of IBM patents*year2003_2004	-.000 (.011)	-.010 (.011)	-.011 (.011)
IBM patents*afteryear2005	-.189** (.086)	-.002 (.100)	-.004 (.097)
square of IBM patents*after year2005	.086** (.010)	.066*** (.010)	.064*** (.011)
IBM opposed patents at EPO	1.886*** (.636)	2.328*** (.765)	2.375*** (.784)
square of IBM opposed patents at EPO	-.544*** (.169)	-.671*** (.191)	-.669*** (.190)
First Stage F-statistic (p-value)	83.63 (.00)	76.40 (.00)	35.17 (.00)
Controls	<i>Sales</i>	<i>Sales, SegmentPatents</i>	<i>Sales, SegmentPatents, PatentThicket</i>

Notes: 1) The full set of IVs include: IBM patents\*year2003\_2004, IBM patents\*afteryear2005, square of IBM patents\*year2003\_2004, square of IBM patents\*afteryear2005, IBM opposed patents in EPO, and square of IBM opposed patents in EPO. 2) The C-test is to assess the exogeneity of IBM patents\*year2003\_2004, IBM patents\*afteryear2005, square of IBM patents\*year2003\_2004, and square of IBM patents\*afteryear2005 as IVs with the null hypothesis that they are valid instruments. 3) The C-test statistic is computed as the difference between two J statistics from GMM estimates: that for the (restricted, fully efficient) regression using the full set of IVs versus that for the (unrestricted, inefficient but consistent) regression using the smaller set of IVs including IBM opposed patents in EPO and square of IBM opposed patents in EPO. 4) The first stage OLS IV Regressions are used as auxiliary regressions to test for weak IVs, as there is no such test in using the GMM estimator. 5) Robust standard errors, clustered by market segment, are in parentheses. 6) \* significant at 10%, \*\* significant at 5%, \*\*\* significant at 1%.

**Table 3.7 GMM Estimates Using Different Sets of IVs**

Dependent variable: OSS entry					
	Specification without interactions		Specification with two-way interactions		
	(1)	(2)	(3)	(4)	(5)
OSS patent pool	.432*** (.163)	.290*** (.133)	-.250 (.295)	-.518** (.217)	-.371 (.321)
cumulativeness			.472 (2.126)	-.315 (2.199)	.479 (2.164)
OSS patent pool * cumulativeness			.361 (.315)	.396* (.261)	.479* (.276)
Concentration			-5.334 (6.275)	-8.291 (6.666)	-7.365 (6.522)
OSS patent pool * Concentration			1.655** (.705)	2.027*** (.564)	1.451*** (.898)
Potential licensors			.779 (1.180)	-1.727 (1.505)	-.788 (2.004)
Sales	-.091 (.317)	-.223 (.204)	-.291 (.255)	-.280 (.225)	-.319 (.227)
Patent quality		1.044 (.675)		2.438 (1.678)	2.322 (1.674)
Total patents				2.625 (1.738)	2.021 (1.745)
SSO patents					-.111 (.129)
Seven year dummies	Yes	Yes	Yes	Yes	Yes
Observations	363	363	363	363	363
Over-identification test statistic (p-value)	8.914 (0.11)	15.691 (0.11)	16.810 (0.10)	17.522 (0.13)	19.096 (0.04)

Notes: 1) The full set of IVs include: IBM patents\*year2003, IBM patents\*afteryear2004, square of IBM patents\*year2003, square of IBM patents\*afteryear2004, IBM opposed patents at EPO, square of IBM opposed patents at EPO. 2) Robustness standard errors, clustered by market segment, are in parentheses. 3) \* significant at 10%, \*\* significant at 5%, \*\*\* significant at 1%. 4) Adding more controls in specification without interactions leads to non-convergence of the GMM estimator.

**Table 3.8 Falsification Test, Impact on Proprietary Software Products**

Dependent variable	New proprietary software product entry		
	(1)	(2)	(3)
OSS patent pool	.007 (.122)	-.011 (.059)	.003 (.065)
Sales	.793 (.505)*	.470 (.530)	.364 (.516)
Total patents		6.841** (2.736)	4.149 (3.095)
Patent quality		7.037* (4.033)	3.793 (4.606)
OIN patents			.034 (.049)
SSO patents			-.116** (.052)
Observations	232	232	232
Log pseudolikelihood	-496.316	-464.505	-451.968

Notes: 1) Robust standard errors, clustered by market segment, are in parentheses. 2) \* significant at 10%, \*\* significant at 5%, \*\*\* significant at 1%. 3) All regressions include market and year fixed effects.

### 3.6.3. Does The Effect Of The OSS Patent Pool Vary With The Patent Thicket?

In this section, I investigate how the impact of the OSS patent pool varies with the cumulateness of innovation and the concentration of patent ownership in a market segment. I begin by examining whether the impact of the OSS patent pool is higher when the cumulateness of innovation in a segment is high. The specification for  $X_{jt}'\beta$  becomes

$$X_{jt}'\beta = \beta_1 OSS\ patent\ pool_{jt} + \beta_2 OSS\ patent\ pool_{jt} * cumulateness_{jt-1} + \gamma_1 Sales_{jt} + \gamma_2 SegmentPatents_{jt-1} + \gamma_3 PatentThicket_{jt-1} + \gamma_4 OtherFreePatents_{jt} + \tau_t. \quad (3.13)$$

To test the direct impact of the OSS patent pool (proposition 1), I present the marginal effect of the OSS patent pool when other variables are at their mean level. The propositions 2 and 3 state how changes in patent pools, cumulateness, and concentration influence the threshold for entry. The testable implications in my data are how changes in each of these variables influence the number of entrants. Thus, if the marginal effect of patent pools on entry is greater when cumulateness is high, this is supportive of proposition 2. Similarly, if the marginal effect of patent pools on entry is greater when concentration is high, this is supportive of proposition 3. To capture the interaction effects as suggested by proposition 2, I compute the marginal effect of the pool when cumulateness is at high and low levels, and test whether the marginal effect of the OSS patent pool is significantly different at these two levels. As before, I employ different specifications by adding the sets of four controls incrementally and use an alternative measure of cumulateness to probe the robustness of the results.

As shown in columns (1)-(3) in table 3.9, a 10% increase in the pool's patent claims is associated with a 2.2%-3% increase in OSS entry, with the effect computed at the average level of cumulateness of innovation. Further, while the marginal effect of the OSS patent pool is insignificant when evaluated at the 10th percentile of the cumulateness, the effects are statistically and economically significant when evaluated at the 90th percentile. Specifically, a 10% increase in the pool's patent claims is

associated with a 3.8%-5.6% increase in OSS entry when cumulateness of innovation is at its 90th percentile. A test for the difference of the two marginal effects (at the 10th and 90th percentiles) is statistically significant. Meanwhile, all the estimates are stable across all specifications and across different measures of cumulateness of innovation.<sup>54</sup> These results suggest that the impact of the OSS patent pool is greater when the cumulateness of innovation is high, providing evidence in support of proposition 2.

The next step is to explore how the impact of the OSS patent pool is influenced by variation in the concentration of patent ownership. The specification can be written as

$$X_{jt}'\beta = \beta_1 OSS\ patent\ pool_{jt} + \beta_3 OSS\ patent\ pool_{jt} * concentration_{jt-1} + \gamma_1 Sales_{jt} + \gamma_2 SegmentPatents_{jt-1} + \gamma_3 PatentThicket_{jt-1} + \gamma_4 OtherFreePatents_{jt} + \tau_t. \quad (3.14)$$

The empirical results for this specification are shown in columns (4)-(6) in table 3.9. While the marginal effect of the OSS patent pool is insignificant when concentration is at its 10th percentile or mean value, a 10% increase in the OSS patent pool's patent claims is associated with a 1.3%-1.7% increase in OSS entry when concentration is at its 90th percentile. Meanwhile, excluding the simplest specification with only sales as control, the test for the difference of marginal effects of the OSS patent pool between concentration evaluated at the 10th percentile and the 90th percentile is statistically significant at the 10% level. Thus, my results provide evidence in support of proposition 3.

To present a more complete picture of how the impact of the OSS patent pool varies with cumulateness of innovation and concentration of patent ownership, I present results including the two sets of interactions together. These estimates are presented in columns (7)-(9) in table 3.9. A 10% increase in the pool's patent claims is associated with a 4.5%-4.8% increase in OSS entry when the cumulateness of innovation is at its 90th

---

<sup>54</sup> I also use a robust measure of cumulateness. The results are very consistent with the baseline measure for the cumulateness.

percentile; the marginal effect of the pool is significantly different at the 1% level when evaluated at high and low levels of cumulateness. While in this specification the interaction between the OSS patent pool and concentration of patent ownership becomes insignificant, the impact of the OSS patent pool is still statistically significant and positive when evaluated at the 90th percentile of concentration and the sign of the interaction's coefficient is positive across specifications. However, there is no statistically significant difference between the marginal effects evaluated at the 10th and 90th percentiles of concentration. Thus, while the qualitative nature of my results is similar when including cumulateness and concentration together, the statistical significance of the concentration result is weaker. This is likely caused by the multicollinearity between the two interactions.<sup>55</sup>

---

<sup>55</sup> In the pooled sample, the simple correlation coefficient between the two interaction terms is 0.66.

**Table 3.9 Conditional Fixed-effect Poisson, Add Two-way Interactions**

Dependent variable: OSS entry	Interaction with Cumulativeness of Innovation			Interaction with Concentration of Patent Ownership			Interaction with Cumulativeness of Innovation and with Concentration of Patent Ownership		
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
OSS patent pool	.079 (.118)	.019 (.103)	.065 (.134)	-.012 (.140)	-.012 (.130)	-.080 (.145)	-.082 (.144)	-.066 (.142)	-.048 (.182)
OSS patent pool * Cumulativeness	.191* (.102)	.265*** (.098)	.314*** (.096)				.297*** (.090)	.289*** (.092)	.290*** (.098)
OSS patent pool * Concentration				.422 (.351)	.541* (.322)	.640* (.381)	.287 (.301)	.310 (.280)	.312 (.386)
Cumulativeness	1.006* (.557)	2.722*** (.758)	3.225*** (1.027)			1.184 (1.019)	3.057*** (.889)	2.934*** (.865)	3.081*** (1.036)
Concentration			-13.901** (6.896)	-9.417 (6.891)	-12.958 (6.628)	-12.632* (7.308)	-14.150*** (5.362)	-14.829*** (5.672)	-15.093** (6.799)
Sales	-.158 (.545)	-.048 (.534)	-.018 (.501)	-.139 (.565)	-.038 (.555)	-.088 (.522)	-.044 (.519)	-.036 (.524)	-.036 (.498)
Potential licensors		0.501 (1.658)	0.965 (1.573)	-.710* (.438)	1.113 (1.645)	1.367 (1.652)	.961 (.643)	1.027 (1.636)	1.038 (1.566)
Total patents		.512 (2.384)	-.317 (2.532)		-3.112 (2.534)	-2.514 (2.610)		-.845 (2.538)	-.700 (2.574)
Patent quality		-.886 (2.333)	-1.176 (2.377)		-1.388 (2.483)	-1.329 (2.418)		-1.183 (2.424)	-1.203 (2.393)
OIN patents			-.059 (.069)			-.067 (.074)			-.062 (.069)
SSO patents			-.020 (.062)			.023 (.079)			-.001 (.075)
Observations	286	286	286	286	286	286	286	286	286
Log pseudolikelihood	-227.205	-225.971	-224.103	-228.567	-227.599	-226.762	-224.319	-224.186	-223.893
<i>Marginal Effects</i>									
OSS patent pool (average)	.221** (.103)	.216** (.094)	.298** (.128)	.081 (.090)	.108 (.087)	.062 (.086)	.202** (.101)	.217** (.101)	.236** (.126)
OSS patent pool (cumulativeness=10%)	.121 (.108)	.077 (.094)	.133 (.129)				.046 (.108)	.066 (.107)	.084 (.124)
OSS patent pool (cumulativeness=90%)	.383*** (.146)	.440*** (.140)	.564*** (.164)				.453*** (.128)	.462*** (.135)	.481*** (.166)
Statistic for the test of the difference between high and low cumulativeness (p-value)	3.49* (0.06)	7.32*** (0.01)	10.71*** (0.00)				10.91*** (0.00)	9.79*** (0.00)	8.82*** (0.00)
OSS patent pool (concentration =10%)				.043 (.107)	.059 (.101)	.003 (.107)	.176 (.110)	.189* (.111)	.207 (.148)
OSS patent pool (concentration =90%)				.129* (.083)	.169** (.081)	.134* (.076)	.234** (.096)	.253*** (.097)	.271** (.109)
Statistic for the test of the difference between high and low concentration (p-value)				1.45 (0.20)	2.83* (0.09)	2.82* (0.09)	0.91 (0.34)	1.23 (0.26)	0.66 (0.40)

Notes: 1) Robust standard errors, clustered by market segment, are in parentheses. 2) \* significant at 10%, \*\* significant at 5%, \*\*\* significant at 1%. 3) The number of observations is lower than in Table 1 because of the use of conditional fixed effects Poisson models, which drop market segments in which there is no new product entry over my sample period. 4) The statistic for the test of the difference of marginal effect of an OSS patent pool between the cumulativeness of innovation/the concentration of patent ownership at the 10th percentile and at the 90th percentile is distributed as chi-square, and the p-value is shown in the parentheses. 5) All regressions include market and year fixed effects.

To evaluate how potential omitted variables may influence my results, I again examine the robustness of my results to the use of instrumental variables. I interact each of the original instruments (for the size of the OSS patent pool) with cumulateness and with patent ownership concentration to form the instruments for the two interaction variables. The estimated results are shown in table 3.10. Columns (1) through (3) provide estimates using the full sample. The estimated coefficients for the two interaction variables remain significantly positive across all specifications. For robustness, the estimates from columns (4) through (6) are based on a sample that excludes market segments without OSS entry during the study period; the coefficients for the two interactions are still significantly positive and stable across all specifications. While the combined empirical evidence from tables 3.9 and 3.10 largely confirms proposition 3, it seems to suggest that the interaction between the OSS patent pool and concentration of patent ownership is not as strong as its interaction with cumulateness.

I also explore the impact of the OSS patent pool when both the cumulateness of innovation and concentration of patent ownership in a segment are high. The detailed empirical results are shown in table 3.11. For segments with both cumulateness of innovation and concentration at high levels (at the 90th percentile), a 10% increase in the pool's patent claims is associated with a 7.5%-8.8% increase of new OSS entry; this marginal effect is significantly greater than that when cumulateness of innovation or concentration (or both) are at lower levels.

**Table 3.10 GMM Estimates with IVs, Add Two-way Interactions**

Dependent variable: OSS entry	Full sample			Sample dropping the segments without OSS entry		
	(1)	(2)	(3)	(4)	(5)	(6)
OSS patent pool	-0.432*** (.130)	-0.679*** (.201)	-0.478 (.341)	-0.401*** (.127)	-1.070*** (.302)	-0.865** (.404)
Cumulativeness	1.956*** (.564)	.587 (1.629)	.697 (2.737)	2.176*** (.592)	.536 (2.110)	1.340 (1.222)
OSS patent pool * Cumulativeness	.567*** (.138)	.605*** (.210)	.583** (.265)	.588*** (.141)	.462* (.256)	.489* (.283)
Concentration	- 15.717** * (4.114)	- 20.963*** (5.950)	-18.632 (6.287)	- 14.483** * (4.133)	- 21.017*** (5.651)	-17.147*** (5.527)
OSS patent pool * Concentration	2.338*** (.417)	2.161*** (.604)	1.626* (.870)	1.976*** (.376)	2.746*** (.626)	1.973* (1.019)
Potential licensors	1.482*** (.355)	-1.036 (1.366)	-0.368 (1.858)	1.551*** (.371)	.291 (1.225)	.888 (1.550)
Sales	-.280 (.261)	-.267 (.245)	-.232 (.242)	-.245 (.277)	-.145 (.247)	-.186 (.232)
Total patents		2.623* (1.547)	1.977 (1.689)		1.285 (1.698)	1.054 (1.861)
Patent quality		4.661*** (1.206)	4.293*** (1.464)		4.669*** (1.138)	4.007*** (1.206)
SSO patents			-.134 (.105)			-.139 (.106)
Year dummies	Yes	Yes	Yes	Yes	Yes	Yes
Observations	363	363	363	286	286	286
Over- identification test statistic (p- value)	19.425 (0.08)	16.201 (0.18)	16.377 (0.18)	19.182 (0.08)	16.622 (0.22)	17.978 (0.16)

Notes: 1) The full set of IVs include: IBM patents\*year2003\_2004, IBM patents\*afteryear2005, square of IBM patents\*year2003\_2004, square of IBM patents\*afteryear2005, IBM opposed patents at EPO, and square of IBM opposed patents at EPO. 2) Robust standard errors, clustered by market segment, are in parentheses. 3) \* significant at 10%, \*\* significant at 5%, \*\*\* significant at 1%. 4) Year dummies include year2001\_2002, year2003, year2004, year2005, year2006, year2007, and year2008\_2009. Including more year dummies would lead to non-convergence of the GMM estimator. 5) I am unable to include OIN patents as a control, as it would also lead to non-convergent estimation results.

**Table 3.11 Conditional Fixed-effect Poisson, Add Two-way/Three-way Interactions**

Dependent variable: OSS entry	(1)	(2)	(3)
OSS patent pool	.232 (.254)	.330 (.264)	.389 (.277)
Cumulativeness	2.741 (2.004)	3.268 (2.280)	3.183 (1.954)
OSS patent pool * Cumulativeness	-.346 (.385)	-.434 (.403)	-.490 (.383)
Concentration	-11.076* (6.869)	-8.843 (6.959)	-10.113 (7.445)
OSS patent pool * Concentration	-1.048 (.831)	-1.360 (.884)	-1.483 (.916)
Cumulativeness *Concentration	-.256 (5.351)	-1.450 (6.006)	-.664 (5.306)
OSS patent pool * cumulateness *Concentration	2.668* (1.502)	3.096** (1.618)	3.318** (1.562)
Potential licensors	.860 (.615)	.001 (1.586)	-.121 (1.516)
Sales	.116 (.518)	.106 (.514)	.115 (.493)
Total patents		-.138 (2.468)	.220 (2.525)
Patent quality		-2.404 (2.658)	-2.464 (2.642)
OIN patents			-.074 (.068)
SSO patents			-.021 (.069)
Observations	286	286	286
Log pseudolikelihood	-222.779	-222.294	-221.829
<i>Marginal Effects</i>			
OSS patent pool (average)	.182* (.103)	.216** (.106)	.242** (.116)
[1] OSS patent pool (cumulativeness=10%, concentration =10%)	.095 (.141)	.146 (.143)	.183 (.155)
[2] OSS patent pool (cumulativeness=10%, concentration =90%)	.001 (.097)	.008 (.097)	.028 (.097)
[3] OSS patent pool (cumulativeness=90%, concentration =10%)	.097 (.252)	.104 (.255)	.104 (.255)
[4] OSS patent pool (cumulativeness=90%, concentration =90%)	.749*** (.169)	.830*** (.199)	.877*** (.200)
Statistic for the test of the difference between [1] and [4] (p-value)	18.96*** (0.00)	16.38*** (0.00)	16.90*** (0.00)
Statistic for the test of the difference between [2] and [4] (p-value)	14.28*** (0.00)	12.88*** (0.00)	13.60*** (0.00)
Statistic for the test of the difference between [3] and [4] (p-value)	3.44* (0.06)	3.76** (0.05)	4.70** (0.03)

Notes: 1) Robust standard errors, clustered by market segment, are in parentheses. 2) \* significant at 10%, \*\* significant at 5%, \*\*\* significant at 1%. 3) The above estimation is based on the specification  $X_{jt} \beta = \beta_1 OSS\ patent\ pool_{jt} + \beta_2 OSS\ patent\ pool_{jt} * cumulateness_{jt-1} + \beta_3 OSS\ patent\ pool_{jt} * concentration_{jt-1} + \delta_1 OSS\ patent\ pool_{jt} * cumulateness_{jt-1} * concentration_{jt-1} + \delta_2 cumulateness_{jt-1} * concentration_{jt-1} + \gamma_1 Sales_{jt} + \gamma_2 Segment\ Patents_{jt-1} + \gamma_3 Patent\ Thicket_{jt-1} + \gamma_4 Other\ Free\ Patents_{jt} + \tau_t$ . 4) The difference between [1] and [2] / between [1] and [3] / between [2] and [3] is insignificant and not included in the table due to the limited space. 5) All regressions include market and year fixed effects.

### 3.7 Conclusions

The empirical evidence demonstrates that increases in the size of an OSS patent pool related to a software segment are associated with increases in OSS entry by start-up software firms in that segment.<sup>56</sup> Furthermore, the impact of the OSS patent pool is magnified when two features of patent thickets are present in the segment: cumulativeness of innovation and concentration of patent ownership. I observe a particularly strong relationship between the size of the patent pool and OSS start-ups' entry in segments with high cumulativeness of innovation; the marginal effect of the pool is also greater when concentration is high, although this result is not robust across all specifications.

In conclusion, my results suggest that OSS patent pools may facilitate markets for technology by strengthening a startup's negotiating position against incumbents with potentially blocking patents. Indeed, by reducing entry costs and the associated incentives for startups to operate in the open source world, OSS patent pools appear to stimulate the open source innovation activities of entrepreneurial firms in industries characterized by dense patent thickets and concentrated property rights.

The study analyzes the impact of patent pools on the behavior of those firms whose entry decisions are most likely to be affected by the change in licensing and negotiation costs: start-up firms considering entry as an OSS competitor. The introduction of OSS patent pools may have secondary implications for two groups of firms that I do not study: large firms and those who sell software under a traditional proprietary license. Understanding the implications of OSS patent pools on these other

---

<sup>56</sup> A natural and interesting extension to this study is to look at how the size of the OSS patent pool influences the survival rate of OSS firms. However, of the firms introducing new OSS products in the sample, only one firm exited before the end of the sampling period. As a result, there is insufficient variance in my data to measure the impact of the OSS patent pool on the survival of start-up firms who produce OSS.

groups will have important implications for the rate and direction of inventive activity in software, and quantifying these implications is an important question for future research.

This study deepens our understanding of the role of patent pools. While prior work has shown that the introduction of traditional patent pools can lead to a decline in innovative activity among both licensors and licensees (Lampe and Moser 2010, Joshi and Nerkar 2011), I find that the introduction of OSS patent pools stimulates the innovative activity of a key group, start-ups. I speculate that this difference may reflect the requirements of use for the OSS patent pool; namely OSS pool patents are offered royalty-free and beneficiaries are required not to sue firms producing OSS. As noted above, the absence of this latter requirement may be one reason for my inability to find a measurable impact of standard setting organizations' patents on OSS entry. However, more research is needed to understand how licensing requirements across the two types of institutions influence innovation outcomes.

**APPENDIX A**

**FOR CHAPTER 2: FURTHER DETAILS ON DATA**

**CONSTRUCTION AND EXAMPLES OF PROJECTS IN THE**

**TREATED GROUPS**

SRDA contains information about each project's main functionality and its intended adopters. This information includes fields such as topics, descriptions, and operating systems. All topics on SRDA are organized in a hierarchical structure. An OSS "topic" in SRDA is defined to be the domain for the set of problems addressed by the OSS. There are 18 top-level topics in total and examples include "Internet," "Communications," and "System." For each top-level topic, there are several levels of sub-topics; for example, under the top-level topic "System," there are second-level topics such as "Operating System Kernels" and "Distributed Computing." Third-level topics are also sometimes provided under a second-level topic. An OSS "description" in SRDA details the OSS project's more specific features; for example, HomePlayer, one of the most popular projects on SourceForge, has a description that reads "HomePlayer is an extension of the FreePlayer software provided by the French Internet provider Free ([www.free.fr](http://www.free.fr)). It adds a lot of functionality like hard-disk browsing, meteo, tv program, etc." Therefore, I expect that the combination of "topic" and "description" will provide me with sufficient information to determine the treatment groups for the difference-in-difference estimation. As I mention in greater detail below, I search both the "topic" and "description" fields for the key words used to determine each treatment group. Last, I also use SRDA's information on "operating system" for projects. An OSS "operating system" describes on which platform the software can run.

*Identifying projects that show technology overlap with the focal litigated technology.* As noted in the body of the text, for the *SCO v. IBM* case I wish to identify Linux kernel projects. To identify such projects, I search the “topic” field in SRDA for the key words “Linux” and “kernel.” One set that largely satisfies this criterion consists of projects with the topic “Linux” under a higher level topic “Operating System Kernels” (i.e., “Linux” projects within the set of “Operating System Kernels” projects). Examples of projects for which “*Linux\_kernel*” is equal to 1 are shown in the below Table A.1. The description field for one such project “TinyLinux” reads “TinyLinux is a small Linux Distribution for i386 derived from SuSE 6.4. In the base version it just contains the things which are necessary to run Linux...”

As noted in the main body of the chapter, for the *FireStar/DataTern v. Red Hat* case I wish to identify projects that use object-relational mapping technology. So I search the “description” field for all OSS projects from SRDA for key words “object,” “relational,” and “mapping.” Then I create a dummy variable “*object\_relational*”: if the project’s “description” field includes these key words, I set the dummy variable “*object\_relational*” to be 1; otherwise I set it to be 0. Examples of projects for which “*object\_relational*” is equal to 1 are shown in Table A.2. The description field of one such project “JGrinder Object/Relational Mapping” notes that “JGrinder is largely an Object to Relational mapping solution for providing Java persistence. It has been used for high volume, high availability solutions.”

*Identifying business projects that are specific to the focal litigated platform.* For the *SCO v. IBM* case, I utilize a two-step method to construct the variable *business\_app\_on\_Linux*. The first step is to identify the set of all types of OSS projects primarily intended for use by firms: I search the “description” field of all OSS projects and identify projects including the key words “enterprise,” “business,” “company,” “ERP,” and “CRM.” I also searched the “topic” field of OSS and use this to identify topics with the key word “point-of-sale.” I identify projects with any of these key words

as enterprise applications. Second, each OSS project from SRDA also provides a field called “operating system” that details the platforms on which a project can run. I identify the set of projects can only run on Linux based on this field. The treated group called “*business\_app\_on\_Linux*” is composed of projects that lie at the intersection of the sets created by these two steps. That is, for projects that lie within this intersection, I set the value of a dummy variable “*business\_app\_on\_Linux*” to be 1; otherwise I set it to be 0. Examples of projects for which “*business\_app\_on\_Linux*” is equal to 1 are shown in Table A.3. The description field of one such project named “gShop” states that “gShop is a complete point of sale application that can be customized to suit most types of businesses.”

For the *FireStar/DataTern v. Red Hat* case, to identify the set of projects specific to JBoss, I search the “description” field of all OSS projects from SRDA for the key word “JBoss.” Accordingly, I create a dummy variable “*JBoss\_related*”: for projects from the set of JBoss-related projects, I set the variable “*JBoss\_related*” to be 1; otherwise I set it to be 0. Examples of projects for which “*JBoss\_related*” is equal to 1 are shown in Table A.4. The description field of one such project “Redpos” states that “A simple and rock solid Point Of Sale (POS) application. The POS is based on the JBoss MicroKernel, has a flexible graphical interface and can easily be connected to different backoffice/ERP systems. Its functionality can be extended during runtime.”

**Table A.1 Examples in the treated group “Linux kernel”**

Project id	Project name	Project Description
5907	Linux/APUS Kernel	Linux Kernel for PowerPC equipped Amiga Computers.
11125	TinyLinux	TinyLinux is a small Linux Distribution for i386 derived from SuSE 6.4. In the base version it just contains the things which are necessary to run Linux. Therefore the base package is rather small and requires approx. 7MB.
33589	system call tracker	A linux kernel module and supporting user space environment which allow interception and modifying system calls that match user defined criteria. Think of it as strace on steroids.

**Table A.2 Examples in the treated group “object relational”**

Project id	Project name	Project Description
1531	JGrinder Object/Relational Mapping	JGrinder is largely an Object to Relational mapping solution for providing Java persistence. It has been used for high volume, high availability solutions.
8706	Osage - Persistence Plus XML	JDBC-based Object-Relational mapping system. It maps Java objects to RDBMS. It generates SQL for retrieving, saving, and deleting objects. It does XML<=>RDBMS. It autogenerates keys, maintains dependent objects and relations.
48132	Cayenne	Cayenne is an object-relational mapping framework written in Java. It provides tools and libraries to implement object persistence on top of relational databases. Cayenne consists of class libraries and a GUI tool for O/R mapping and deployment.

**Table A.3 Examples in the treated group “*business app on Linux*”**

Project id	Project name	Project Description
8854	VPrice	VPrice is a cost price calculator for Linux. It is designed for producers who need to have their product prices calculated “on the fly”.
10974	LnxFire	LnxFire is a Linux & Gnome firewall tool for the small business owner. Use the firewall creation wizard to quickly create a basic firewall. Featuring dynamic rule modifiers, proactive monitoring, reporting, email alerts and auto lockout of port scanners.
16639	gShop	gShop is a complete point of sale application that can be customized to suit most types of businesses.

**Table A.4 Examples in the treated group “*JBoss related*”**

Project id	Project name	Project Description
46485	JBoss JBuilder OpenTool	Voyager JBoss OpenTool is an OT for JBuilder 6 Enterprise Edition. It integrates the JBoss application server the same way other application server are already integrated. Development and support was stopped since JBuilderX comes with JBoss support now.
67328	JBoss Integration Plugin for IDEA	This plugin gives us the ability to integrate JBoss 3.0.x with Jetty 4.x into IDEA. The objective is to have an easy way to test and deploy web apps. Future versions of the plugin will support the combo of JBoss/Tomcat.
104726	Redpos	A simple and rock solid Point Of Sale (POS) application. The POS is based on the JBoss MicroKernel, has a flexible graphical interface and can easily be connected to different backoffice/ERP systems. Its functionality can be extended during runtime.

## APPENDIX B

### FOR CHAPTER 3: PROOFS IN SECTION 3.2

#### Proof for Lemma 1

As assumed,  $\Delta$  is distributed uniformly and  $g(\Delta) = 1$ , so  $G(\tilde{\Delta}) = \tilde{\Delta}$ ,  $G(\Delta_s(T)) = \Delta_s(T)$ . Given  $q(\Delta, b) = b(1 - \Delta)$ , I have:

$$\begin{aligned} \max_T \pi(v, b, L | \tilde{\Delta}) &= \max_T \int_{\Delta_s(T)}^1 (b(1 - \Delta)v - L) \frac{1}{1 - \tilde{\Delta}} d\Delta + T \cdot \frac{\Delta_s(T) - \tilde{\Delta}}{1 - \tilde{\Delta}} \\ &= \max_T \frac{1}{1 - \tilde{\Delta}} \int_{\Delta_s(T)}^1 (bv - bv\Delta - L) d\Delta + T \cdot \frac{\Delta_s(T) - \tilde{\Delta}}{1 - \tilde{\Delta}} \end{aligned}$$

Let's denote  $\Delta_s(T)$  as  $\Delta_s$  and since it is more convenient we compute the first order condition with respect to  $\Delta_s$ ,

$$\begin{aligned} \max_{\Delta_s} \pi(v, b, L | \tilde{\Delta}) &= \max_{\Delta_s} \frac{1}{1 - \tilde{\Delta}} \left( bv - \frac{1}{2}bv - L - bv\Delta_s + \frac{\Delta_s^2}{2}bv + \Delta_s L \right) + \frac{\Delta_s - \tilde{\Delta}}{1 - \tilde{\Delta}} \cdot \\ & (b(1 - \Delta_s)v + L) \end{aligned}$$

$$\frac{\partial \pi}{\partial \Delta_s} = \frac{1}{1 - \tilde{\Delta}} (-bv + \Delta_s bv + L) + \frac{b(1 - \Delta_s)v + L}{1 - \tilde{\Delta}} + \frac{\Delta_s - \tilde{\Delta}}{1 - \tilde{\Delta}} \cdot (-vb)$$

The first order condition  $\frac{\partial \pi}{\partial \Delta_s} = 0$  is equal to  $-\Delta_s bv + 2L + vb\tilde{\Delta} = 0$ .

That is, the optimal  $\Delta^s$  for a given correct belief  $\tilde{\Delta}$  is:

$$\Delta^{s*} = \frac{1}{bv} (2L + vb\tilde{\Delta}) \tag{B.1}$$

Since  $T = b(1 - \Delta_s)v + L$ , the corresponding  $T$  for a given correct belief  $\tilde{\Delta}$  is

$$T = bv - (2L + vb\tilde{\Delta}) + L \tag{B.2}$$

The unique sequential equilibrium in pure strategies can be obtained from the following two conditions, where the first condition (equation (B.3)) describes the optimal  $T(v, b, L | \tilde{\Delta})$  given a  $\tilde{\Delta}$  and the second condition (equation (B.4)) describes how the decision of OSS entry is determined by a given  $T$ :

$$T^* = bv - (2L + vb\tilde{\Delta}^*) + L \tag{B.3}$$

$$\tilde{\Delta}^* = T^* + c - v \tag{B.4}$$

Combining these two conditions, we have

$$\tilde{\Delta}^* = vb - (2L + vb\tilde{\Delta}^*) + L + c - v$$

Therefore, the optimal improvement level  $\tilde{\Delta}^*$  for entry has the following form:

$$\underline{\Delta}^* = (1 + vb)^{-1}(vb + c - L - v) \quad (\text{B.5})$$

Further, to satisfy  $\underline{\Delta}^* \in [0,1]$ :

$$vb + c - L - v < 1 + vb, \text{ i.e. } L + v + 1 - c > 0 \quad (\text{B.6})$$

### Proof for Proposition 1

$$\begin{aligned} \frac{\partial \underline{\Delta}^*}{\partial b} &= -(1 + vb)^{-2}v(vb + c - L - v) + (1 + vb)^{-1}v \\ &= v(1 + vb)^{-2}(1 + vb - vb - c + L + v) = v(1 + vb)^{-2}(L + v - c + 1) \end{aligned}$$

As shown in equation (B.6) that  $L + v + 1 - c > 0$ ,  $\frac{\partial \underline{\Delta}^*}{\partial b} > 0$ .

Further, since  $\frac{\partial b}{\partial x} < 0$ , based on the chain rule of computing derivatives, we

immediately have  $\frac{\partial \underline{\Delta}^*}{\partial x} = \frac{\partial \underline{\Delta}^*}{\partial b} \frac{\partial b}{\partial x} < 0$ .

### Proof for Lemma 2

Since  $\frac{\partial \underline{\Delta}^*}{\partial x} = \frac{\partial \underline{\Delta}^*}{\partial b} \frac{\partial b}{\partial x} = \frac{\partial b}{\partial x} v(1 + vb)^{-2}(L + v - c + 1)$ ,

$$\begin{aligned} \frac{\partial^2 \underline{\Delta}^*}{\partial x \partial v} &= \frac{\partial b}{\partial x} [-2(1 + vb)^{-3}b(vL + v^2 - vc + v) + (1 + vb)^{-2}(L + 2v - c + 1)] \\ &= \frac{\partial b}{\partial x} (1 + vb)^{-3} [(1 + vb)(L + 2v - c + 1) - 2b(vL + v^2 - vc + v)] \end{aligned}$$

That is, to show  $\frac{\partial^2 \underline{\Delta}^*}{\partial x \partial v} < 0$ , because  $\frac{\partial b}{\partial x} < 0$ , it remains to prove

$$(1 + vb)(L + 2v - c + 1) - 2b(vL + v^2 - vc + v) > 0$$

That is, to prove  $L + 2v - c + 1 + vbL + 2bv^2 - vbc + vb - 2bvL - 2bv^2 + 2bvc - 2bv > 0$

That is, to prove  $L + 1 + v + v + bvc > c + bv + bvL$

Since it has been shown  $L + 1 + v > c$ , it remains to show  $v + bvc > bv + bvL$

First, since  $b < 1$ ,  $v > bv$ . Second, in equation (B.5), it needs to satisfy  $vb + c - L - v > 0$ . Since  $vb < v$ ,  $c$  needs to be larger than  $L$ , and therefore  $bvc > bvL$ . So  $v + bvc > bv + bvL$ .

### Proof for Proposition 2

From the Lemma 2, it has been shown  $\frac{\partial^2 \underline{\Delta}^*}{\partial x \partial v} < 0$ . Thus,  $\frac{\partial^2 \underline{\Delta}^*}{\partial x \partial \theta} = \frac{\partial^2 \underline{\Delta}^*}{\partial x \partial v} \cdot \frac{\partial v}{\partial \theta} > 0$ , as  $\frac{\partial v}{\partial \theta} = -V(1 - \frac{1}{n}) < 0$ .  $\frac{\partial^2 \underline{\Delta}^*}{\partial x \partial \theta} > 0$  suggests that the impact of the OSS patent pool on the threshold for new OSS product entry  $\underline{\Delta}^*$  (OSS patent pool size  $x$  reduces this threshold) will be *lower* when  $\theta$  is *high*. Therefore, on the other hand, the impact of the OSS patent pool on the threshold for new OSS product entry  $\underline{\Delta}^*$  will be *higher* when the cumulativeness of innovation is high (i.e. when  $\theta$  is *low*).

### Proof for Proposition 3

From the Lemma 2, it has been shown  $\frac{\partial^2 \underline{\Delta}^*}{\partial x \partial v} < 0$ . Thus,  $\frac{\partial^2 \underline{\Delta}^*}{\partial x \partial n} = \frac{\partial^2 \underline{\Delta}^*}{\partial x \partial v} \cdot \frac{\partial v}{\partial n} > 0$ , as  $\frac{\partial v}{\partial n} = -V\theta n^{-2} < 0$ .  $\frac{\partial^2 \underline{\Delta}^*}{\partial x \partial n} > 0$  suggests that the impact of the OSS patent pool on the threshold for new OSS product entry  $\underline{\Delta}^*$  (OSS patent pool size  $x$  reduces this threshold) will be *lower* when  $n$  is *high*. Therefore, on the other hand, the impact of the OSS patent pool on the threshold for new OSS product entry  $\underline{\Delta}^*$  will be *higher* when the concentration of patent ownership is high (i.e. when  $n$  is *low*).

## APPENDIX C

### FOR CHAPTER 3: IDENTIFICATION OF SOFTWARE SEGMENTS AND THE MATCHING PATENT CLASSES

#### C.1 Identify Software Segments

To measure entry with new OSS products related to different software segments in each year, a crucial step is to divide the software market into different segments that are reasonably distinct from each other. One main source of software segments is the product code classification system embedded in the PROMT database. For a portion of news articles from PROMT, there are a few product codes assigned to each new article that indicate what product category/categories are associated with that article. All these product categories are organized as a hierarchical structure by PROMT and are defined both in terms of customer segments and technologies. Table C.1 shows some examples of PROMT codes.

However, there are two drawbacks to just relying on PROMT classifications. First, a significant percentage (about 60%) of OSS product introduction news articles from PROMT is missing the product code field. Thus, I must manually assign product codes for this set of articles. Second, the PROMT classes do not include keywords, making it difficult to manually match articles to PROMT classes. Thus, I further match PROMT product code classes with CorpTech product code classes<sup>57</sup> to take advantage of the keywords defined for each CorpTech product code. The resulting concordance table (denoted as the PROMT-CorpTech concordance hereafter) consists of about 80 PROMT

---

<sup>57</sup> There are more than 290 software product codes (denoted as SOF) defined by CorpTech Directory. Each firm in this directory is associated with a set of self-reported product codes selected from these 290 SOF categories.

codes matched to CorpTech’s six-digit or seven-digit product codes. Each product code is associated with a set of technology phrases specific to that product code. This is used as a basis for me to identify (i) the PROMT articles with missing product codes and (ii) the related patents across a variety of software segments. Table C.2 shows some examples of the PROMT-CorpTech concordance.

### **C.2 Identify Patent Classes across Software Segments**

Using the NBER patent data project and USPTO database, I constructed my patent dataset, which consists of all patents granted from 1976 to 2009. The sample period is from 1999 to 2009. To identify the related patents across a range of market segments from the PROMT-CorpTech concordance, I first examined specialist firms that produce in only one software segment and particularly only one CorpTech six- or seven-digit code<sup>58</sup>. The sample of single specialists is from the CorpTech directory, over 1992 to 2004 and 2010.<sup>59</sup> I found 3500 patents held by about 700 specialists that operate in different software markets from the PROMT-CorpTech concordance. The 3-digit USPTO classes to which the 3500 patents and their forward citations belong served as a starting point for me to map patent classes to each product code: for each product code, the top decile of these 3-digit US classes was used as candidates representing the core technologies for that code.

While the procedure I use is similar to the one used by Cockburn and MacGarvie (CM) (2011), I constructed my own classification for several reasons. First, my sample period is more recent than theirs, so the mapping between patent technologies and product markets may have changed over time. Second, Cockburn and MacGarvie examined 25 specific product codes that have incomplete overlap with the open source

---

<sup>58</sup> Examples of CorpTech code are provided in Table C.2.

<sup>59</sup> Unfortunately, data from the CorpTech Directory from 2005 to 2009 was not available.

product market segments that I study. However, I did find many similarities between their classification and ours: for the product codes in both their and my classifications, the corresponding patent classes are very similar. Finally, I took the intersection of the patent classes from the patents in the OSS patent pool with the above mapping, which lead to 34 US patent classes and their corresponding product codes.

### **C.3 Match Software Segments with Patent Class-subclass Combinations**

Because most of the 3-digit US patent classes contain quite heterogeneous technologies, I then further generated a more detailed mapping between software product codes and US patent subclass levels by searching for technology phrases associated with each product code. This process generated the final mapping between software segments and patent class-subclass combinations. I further consolidated all product codes into 33 software segments based on whether they are supported by the same technologies (similar patent classes), as I am most interested in whether the supply of certain technologies by the OSS patent pools helps start-ups move into new technology area. The final concordance that I used in the empirical analyses consists of 33 software segments matched to 422 patent class-subclass combinations. Table C.3 shows some examples of this final concordance between software segments and US patent class-subclass combinations. Figures C.1 and C.2 present a more concrete view on the above three steps.

**Table C.1 Examples of PROMT Codes**

---

7372502	Operating systems
7372503	Operating system enhancements
7372504	Graphical user interface software
7372505	Portable document software
7372510	Software development tools
7372511	CASE software
7372512	Programming utilities
7372513	Application development software
7372514	Debugging & testing software
7372520	Peripheral support software
7372521	Device driver software
7372522	Data acquisition software
7372523	Printer support software
7372530	Disk/file management software

---

**Table C.2 Examples of the PROMT-CorpTech Concordance**

---

CorpTech Code	PROMT Product Code
SOF-CS-F	7372650 Fax software
SOF-DM-M	7372421 DBMS
SOF-HL-M	7372466 Medical practice software
SOF-ME-S	7372544 Sound/audio software
SOF-OA-MB	7372662 BBS software
SOF-OA-MC	7372674 Videoconferencing software
SOF-OA-ME	7372605 Electronic mail software
SOF-OA-MG	7372630 Workgroup software
SOF-OA-P	7372441 DTP software
SOF-TS-EC	7372433 Civil engineering software
SOF-TS-EE	7372434 Electrical engineering software
SOF-TS-ER	7372423 Geographic information systems
SOF-UT-H	7372521 Device driver software
SOF-UT-O	7372561 Data center management software
SOF-UT-Q	7372513 Application development software
SOF-UT-X	7372691 Data encryption software

---

**Table C.3 Examples of the Concordance between Segments and Patent Classes**

Software Segment	US class	Subclass Level 0	Subclass Level 1
Artificial Intelligence Software	706	Fuzzy Logic Hardware	Fuzzy Neural Network
Artificial Intelligence Software	706	Knowledge Processing System	Creation Or Modification
Artificial Intelligence Software	706	Knowledge Processing System	Knowledge Representation And Reasoning Technique
Artificial Intelligence Software	706	Neural Network	Learning Method
Artificial Intelligence Software	706	Neural Network	Learning Task
Artificial Intelligence Software	706	Neural Network	Neural Simulation Environment
Artificial Intelligence Software	706	Neural Network	Structure
Artificial Intelligence Software	706	Plural Processing Systems	
Data Encryption Software	380	Communication System Using Cryptography	Having Compression
Data Encryption Software	380	Communication System Using Cryptography	Time Segment Interchange
Data Encryption Software	380	Facsimile Cryptography	Including Generation Of An Associated Coded Record
Data Encryption Software	380	Key Management	Having Particular Key Generator
Data Encryption Software	380	Key Management	Key Distribution
Data Encryption Software	380	Particular Algorithmic Function Encoding	NBS/DES Algorithm
Data Encryption Software	380	Particular Algorithmic Function Encoding	Public Key
Data Encryption Software	380	Video Cryptography	Copy Protection Or Prevention
Data Encryption Software	726	Access Control Or Authentication	Network
Data Encryption Software	726	Access Control Or Authentication	Stand-Alone
Data Encryption Software	726	Monitoring Or Scanning Of Software Or Data	
Data Encryption Software	726	Including Attack Prevention	Intrusion Detection
Data Encryption Software	726	Protection Of Hardware	Theft Prevention

Note: 1) US patent class 706 is described as “Data processing: artificial intelligence”; US patent class 380 is described as “Cryptography”; US patent class 726 is described as “Information security”. 2) All subclasses within each US patent class are structured hierarchically. “Subclass level 0” means the subclass is on the highest level and “Subclass level 1” means the subclass is on the second highest level. My mapping is based on subclass level 1.

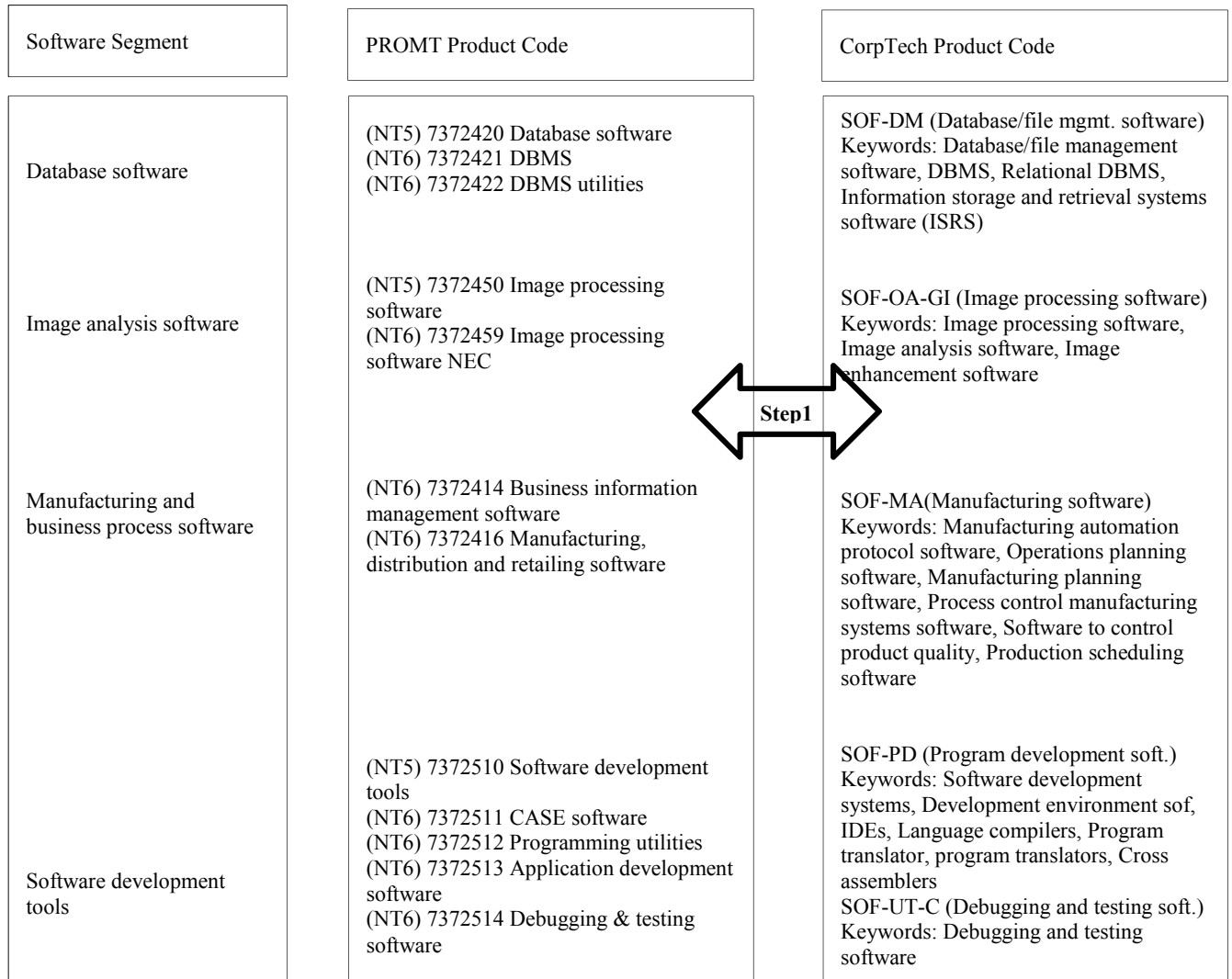


Figure C.1 Identification of Software Segments

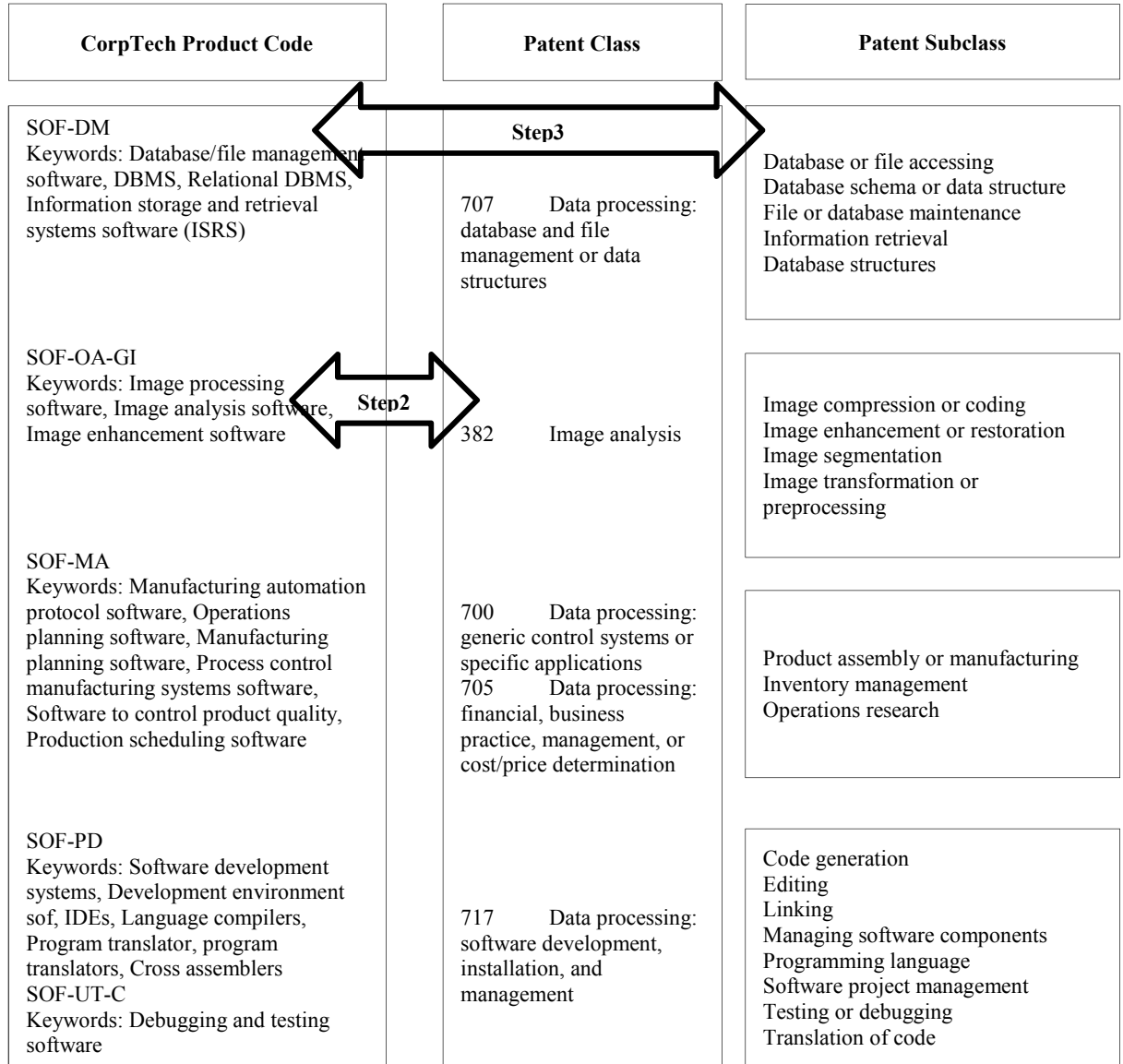


Figure C.2 Mapping Software Segments to Patent Subclasses

## **APPENDIX D**

### **FOR CHAPTER 3: KEYWORDS USED TO IDENTIFY OSS ENTRY**

I used the following set of keywords to search in PROMT news articles for introduction of software products that are licensed as open source. A software product is tagged as open source if it contains any of these keywords. I first implement automatic search and then manually check the results to ensure it is licensed as open source. The choice of open source license terms is based on the distribution of open source licenses used by OSS projects at SourceForge.net, which is the largest repository of OSS. Over 230,000 projects and over 3 million users and developers were registered before the end of year 2009 (SourceForge 2009).

#### **D.1 Keywords related to generic terms of OSS**

open source , open-sourced, OSS, FLOSS, source code, GPL-compatible, non-copyleft, copyleft, free software license, open source license, open-source license, public domain.

#### **D.2 Keywords related to open source licenses**

GPL, General Public License , GNU, Lesser General Public License, LGPL, BSD, FreeBSD, Apache License, Apache Software License, Artistic License, MIT License, Mozilla Public License.

## REFERENCES

- Alexy, O., Reitzig, M. G. 2011. Private-Collective Innovation, Competition, and Firms' Counterintuitive Appropriation Strategies. Available at SSRN: <http://ssrn.com/abstract=1430328>
- Angrist, J.D., J.S. Pischke. 2009. *Mostly harmless econometrics: An empiricist's companion*. Princeton University Press, Princeton, NJ.
- Arora, A, A. Fosfuri, and A. Gambardella. *Markets for technology: The economics of innovation and corporate strategy*. Cambridge: MIT Press.
- Asay, Matt. 2010. What Apple's and Microsoft's patent threats mean for start-ups. CNET. Available at [http://news.cnet.com/8301-13505\\_3-10466670-16.html](http://news.cnet.com/8301-13505_3-10466670-16.html)
- Bertrand, M., E. Duflo, S. Mullainathan. 2004. How much should we trust differences-in-differences estimates? *Quarterly Journal of Economics*. 119(1) 249-275.
- Bessen, J. E. and R. M. Hunt. 2007. An empirical look at software patents. *Journal of Economics & Management Strategy* 16(1): 157-89.
- Bessen, J., E. Maskin. 2009. Sequential innovation, patents, and imitation. *Rand Journal of Economics*. 40(4) 611-635.
- Bhattacharjee, S., R. Gopal, K. Lertwachara, J. Marsden. 2006. Impact of legal threats on online music sharing activity: An analysis of music industry legal actions. *The Journal of Law and Economics*. 49(1) 91-114.
- Bonaccorsi, A., S. Giannangeli, C. Rossi. 2006. Entry strategies under competing standards: Hybrid business models in the open source software industry. *Management Science*. 52(7): 1085-1098.
- Bresnahan, T.F., S. Greenstein. 1996. Technical progress and co-invention in computing and in the uses of computers. *Brookings Papers on Economic Activity: Microeconomics* (1996): 1-83.

- Bresnahan, T.F., S. Greenstein. 1999. Technological competition and the structure of the computer industry. *Journal of Industrial Economics*. 47(1) 1-40.
- Broersma, M. 2007. IDC: Open-Source Market to Be Worth \$5.8B by 2011. CIO. Available at [http://www.cio.com/article/116201/IDC\\_Open\\_Source\\_Market\\_to\\_Be\\_Worth\\_5.8B\\_by\\_2011](http://www.cio.com/article/116201/IDC_Open_Source_Market_to_Be_Worth_5.8B_by_2011)
- Caballero, R.J., A.B. Jaffe. 1993. How high are the giants' shoulders: An empirical assessment of knowledge spillovers and creative destruction in a model of economic growth. In O. Blanchard and S. Fischer, eds., *National Bureau of Economic Research Macroeconomics Annual*, Vol. 8, pp. 15-86, MIT Press, Cambridge, MA.
- Cameron A.C., P. K. Trivedi. 2005. *Microeconometrics: Methods and applications*. Cambridge University Press.
- Chandrashekar, M., R. Mehta, R. Chandrashekar, R. Grewal. 1999. Market motives, distinctive capabilities, and domestic inertia: A hybrid model of innovation generation. *Journal of Marketing Research*. 95-112.
- Chengalur-Smith I., A. Sidorova, S. Daniel. 2010. Sustainability of free/libre open source projects: A longitudinal study. *Journal of the Association for Information Systems*. 11(11): 657-683.
- Clarkson, G. 2005. Patent Informatics for Patent Thicket Detection: A Network Analytic Approach for Measuring the Density of Patent Space. *Working Paper*. Available at: <http://w4.stern.nyu.edu/emplibrary/ACFItbnmV.pdf>
- Cockburn, I. M., M. MacGarvie. 2006. Entry, Exit and Patenting in the Software Industry. NBER Working paper 12563, National Bureau of Economic Research, Cambridge, MA.
- Cockburn, I. M., M. MacGarvie. 2009. Patents, thickets and the financing of early-stage firms: Evidence from the software industry. *Journal of Economics & Management Strategy*. 18(3): 729-773.
- Cockburn, I.M., M.J. MacGarvie. 2011. Entry and patenting in the software industry. *Management Science*. 57(5) 915-933.

- Cohen, J.E., M.A. Lemley. 2001. Patent scope and innovation in the software industry. *California Law Review*. 89(1).
- Comino S., F. M. Manenti, M. L. Parisi, 2007. From planning to mature: On the success of open source projects. *Research Policy*. 36 (10): 1575-1586.
- Crowston, K., H. Annabi, J. Howison. 2003. Defining open source software project success. *Proceedings of International Conference on Information Systems*. Atlanta, GA.
- Dahlander, L. 2007. Penguin in a new suit: a tale of how de novo entrants emerged to harness free and open source software communities. *Industrial and Corporate Change*. 16 (5): 913-943.
- Dasgupta, P., P.A. David. 1994. Toward a new economics of science. *Research Policy*. 23(5): 487-521.
- Dillon, M. 2008. Firestar. Web log comment. Available at <http://blogs.oracle.com/dillon/entry/firestar> (Accessed December 12, 2011).
- Farrell, J., R.P. Merges. 2004. Incentives to challenge and defend patents: Why litigation won't reliably fix patent office errors and why administrative patent review might help. *Berkeley Technology Law Journal*. 19(3) 943-970.
- Farrell, J., P. Klemperer. 2007. Coordination and lock-in: Competition with switching costs and network effects. In M. Armstrong, R. H. Porter, eds., *Handbook of Industrial Economics*, Vol. 3, (Amsterdam: Elsevier, pp. 1967-2072.
- Feller, J., B. Fitzgerald, S.A. Hissam, K.R. Lakhani. 2005. *Perspectives on Free and Open Source Software*. MIT Press, Cambridge.
- Fleming, L. 2001. Recombinant uncertainty in technological search. *Management Science*. 47(1): 117-132.
- Forman, C. 2005. The corporate digital divide: Determinants of Internet adoption. *Management Science*. 51(4) 641-654.

- Forman, C., A. Ghose, A. Goldfarb. 2009. Competition between local and electronic markets: How the benefit of buying online depends on where you live. *Management Science*. 55(1) 47-57.
- Fosfuri, A., M. S. Giarratana, A. Luzzi. 2008. The penguin has entered the building: The commercialization of open source software products. *Organization Science*. 19(2): 292-305.
- Galasso, A., M. Schankerman. 2010. Patent thickets, courts, and the market for innovation. *RAND Journal of Economics*. 41(3): 472-503.
- Gallant, A.R. 1987. *Nonlinear Statistical Models*. John Wiley & Sons, Inc.
- Gambardella, A., B. H. Hall. 2006. Proprietary versus public domain licensing of software and research products. *Research Policy*. 35(6): 875-892.
- Gillen, A. 2009. Linux in the mainstream: Growing deployment of business-critical workloads. IDC Research Report. Accessed at [ftp://ftp.software.ibm.com/linux/pdfs/IDC-Business\\_Critical\\_Workloads\\_on\\_Linux.pdf](ftp://ftp.software.ibm.com/linux/pdfs/IDC-Business_Critical_Workloads_on_Linux.pdf) on May 24, 2012.
- Goetsch, K. 2004. *SCO Group v. IBM: The future of open-source software*. *Journal of Law, Technology, & Policy*. 2003(2) 581-588.
- Govindaraj, S., P. Lee, D. Tinkelman. 2007. Using the event study methodology to measure the social costs of litigation - re-examination using cases from the automobile industry. *Review of Law and Economics*. 3(2) 1-42.
- Graham, S.J.H., D.C. Mowery. 2003. Intellectual property protection in the US software industry. *Patents in the Knowledge-based Economy*. National Academies Press, Washington, D.C.
- Graham, S.J.H., D.C. Mowery. 2005. The use of USPTO "continuation" applications in the patenting of software: Implications for free and open source. *Law & Policy*. 27(1) 128-151.
- Grewal, R., G.L. Lilien, G. Mallapragada. 2006. Location, location, location: How network embeddedness affects project success in open source systems. *Management Science*. 52(7) 1043.

- Griliches, Z. 1990. Patent statistics as economic indicators: a survey. *Journal of Economic Literature*. 27: 1661–1707.
- Haefliger, S., G. Von Krogh, S. Spaeth. 2008. Code reuse in open source software. *Management Science*. 54(1) 180-193.
- Hahn, J., J.Y. Moon, C. Zhang. 2008. Emergence of new project teams from open source software developer networks: Impact of prior collaboration ties. *Information Systems Research*. 19(3) 369-391.
- Hall, B.H., Helmers, C. 2011. Innovation and Diffusion of Clean/Green Technology: Can Patent Commons Help? NBER Working Paper 16920, National Bureau of Economic Research, Cambridge, MA.
- Hall, B.H., S. Graham, D. Harhoff. 2003. Prospects for Improving Us Patent Quality Via Post-Grant Opposition. NBER Working Paper 9731, National Bureau of Economic Research, Cambridge, MA.
- Hall B.H., A. Jaffe, M. Trajtenberg. 2005. Market value and patent citations. *RAND Journal of Economics*. 36 (1): 16 – 38.
- Hall, B. H., A. Jaffe, M. Trajtenberg. 2001. The NBER Patent Citation Data File: Lessons, Insights and Methodological Tools. NBER Working Paper 8498, National Bureau of Economic Research, Cambridge, MA.
- Hann, I.H., J. Roberts, S. Slaughter, R. Fielding. 2004. An empirical analysis of economic returns to open source participation. Faculty Development Grant und Carnegie Bosch Institute, Carnegie Mellon University.
- Heller, M.A., R.S. Eisenberg. 1998. Can patents deter innovation? The anticommons in biomedical research. *Science*. 280(5364) 698-701.
- Henkel, J. 2006. Selective revealing in open innovation processes: The case of embedded Linux. *Research Policy*. 35(7) 953-969.
- Herbsleb, J.D., A. Mockus. 2003. An empirical study of speed and communication in globally distributed software development. *IEEE Transactions on Software Engineering*. 29(6): 481-494.

- Hertel, G., S. Niedner, S. Herrmann. 2003. Motivation of software developers in open source projects: an Internet-based survey of contributors to the Linux kernel. *Research Policy*. 32(7) 1159-1177.
- Heubner, R.A., G. Oancea, R.P. Donald, J.E. Coleman. 2000. Object model mapping and runtime engine for employing relational database with object oriented software. USPTO.
- Jaffe, A.B, M.S. Fogarty, B.A. Banks. 1998. Evidence from Patents and Patent Citations on the Impact of NASA and Other Federal Labs on Commercial Innovation. *Journal of Industrial Economics*. 46(2): 183-205.
- Jaffe A., M. Trajtenberg, R. Henderson. 1993. Geographic localization of knowledge spillovers as evidenced by patent citations. *The Quarterly Journal of Economics*. 108(3): 577-598.
- Joshi, A.M., A. Nerkar. 2011. When Do Strategic Alliances Inhibit Innovation by Firms? Evidence from Patent Pools in the Global Optical Disc Industry. *Strategic Management Journal*. 32(11): 1139-1160.
- Kim, J., G. Marschke. 2005. Labor Mobility of Scientists, Technological Diffusion, and the Firm's Patenting Decision. *RAND Journal of Economics*. 36(2): 298-317.
- Krishnamurthy, S.. 2002. Cave or community? An empirical examination of 100 mature open source projects. Bothell, WA: University of Washington, Bothell.
- Krishnan, A. 2006. New company wasn't all Red Hat acquired. News & Observer. McClatchy-Tribune News Service, Raleigh.
- Lakhani, K.R., E. Von Hippel. 2003. How open source software works: "free" user-to-user assistance. *Research Policy*. 32(6) 923-943.
- Lakhani, K.R., R. G. Wolf. 2003. Why hackers do what they do: Understanding motivation and effort in free/open source software projects. In J. Feller, B. Fitzgerald, S. Hissam, K. R. Lakhani, eds., *Perspectives on Free and Open Source Software*. Cambridge: MIT Press, pp. 3-21.

- Lampe, R., P. Moser. 2010. Do Patent Pools Encourage Innovation? Evidence from the Nineteenth-Century Sewing Machine Industry. *The Journal of Economic History*. 70(4): 898-920.
- Lanjouw, J.O., J. Lerner. 2001. Tilting the table? The use of preliminary injunctions. *The Journal of Law and Economics*. 44(2) 573-603.
- Lanjouw, J.O., M. Schankerman. 2001. Characteristics of patent litigation: a window on competition. *Rand Journal of Economics*. 32(1) 129-151.
- Layne-Farrar, A., J. Lerner. 2010. To join or not to join: Examining patent pool participation and rent sharing rules. *International Journal of Industrial Organization*. 29(2) 294-303.
- Lerner, J., J. Tirole. 2002. Some simple economics of open source. *Journal of Industrial Economics*. 50(2): 197-234.
- Lerner, J., J. Tirole. 2004. Efficient patent pools. *American Economic Review*. 94(3): 691-711.
- Lerner, J., J. Tirole. 2005a. The economics of technology sharing: Open source and beyond. *Journal of Economic Perspectives*. 19(2) 99-120.
- Lerner, J., J. Tirole. 2005b. The scope of open source licensing. *Journal of Law, Economics, and Organization*. 21(1) 20-56.
- Lerner, J., M. Schankerman. 2010. *The Comingled Code: Open Source and Economic Development*. Cambridge: MIT Press.
- Lerner, J., M. Strojwas, J. Tirole. 2007. The design of patent pools: The determinants of licensing rules. *The RAND Journal of Economics*. 38(3): 610-25.
- Lerner, J., F. Zhu. 2007. What Is the Impact of Software Patent Shifts? Evidence from Lotus V. Borland. *International Journal of Industrial Organization*. 25(3): 511-29.
- Lévêque, F., Ménière, Y. 2007. Copyright versus Patents: The Open Source Software Legal Battle. *Review of Economic Research on Copyright issues*. 4(1): 27-46.

- Llobet, G. 2003. Patent litigation when innovation is cumulative. *International Journal of Industrial Organization*. 21: 1135-1157.
- Maurer, S.M., S. Scotchmer. 2006. Open source software: the new intellectual property paradigm. Chapter 5. *Handbooks in Information Systems*. Elsevier B.V., Oxford.
- McMillan, R. 2003. Open source confronts IP issues. InfoWorld. Available at. <http://www.infoworld.com/t/platforms/open-source-confronts-ip-issues-701>.
- Merges, R. P., R. R. Nelson. 1990. On the Complex Economics of Patent Scope. *Columbia Law Review*. 90(4) 839-916.
- Merton, R. K. 1957. Priorities in scientific discovery: A chapter in the sociology of science. *American Sociological Review*. 22(6): 635-659.
- Mishra, B., Prasad, A., Raghunathan, S. 2002. Quality and profits under open source versus closed source. In *Proceedings of the Twenty-third International Conference on Information Systems*. 349-363
- Noel, M.D., M.A. Schankerman. 2006. Strategic Patenting and Software Innovation: Theory and evidence from a panel of U.S. firms. CEPR Working Paper 5701, Center for Economic Policy Research, Washington, DC.
- Muelle, F. 2011. Red Hat feeds the patent trolls and fools the FOSS community. Available at <http://www.fosspatents.com/2011/03/red-hat-feeds-patent-trolls-and-fools.html>
- Murray, F., S. Stern. 2007. Do formal intellectual property rights hinder the free flow of scientific knowledge? An empirical test of the anti-commons hypothesis. *Journal of Economic Behavior and Organization*. 63(4) 648-687.
- Murray, F., S. Stern. 2008. Learning to live with patents: A dynamic model of a knowledge community's response to legal institutional change. NBER working paper.
- Nelson, R., S. Winter. 1982. *An Evolutionary Theory of Economic Change*. Cambridge: Belknap Press.

- Netcraft. 2009. October 2009 Web server survey. Available at.  
[http://news.netcraft.com/archives/2009/10/17/october\\_2009\\_web\\_server\\_survey.html](http://news.netcraft.com/archives/2009/10/17/october_2009_web_server_survey.html).
- O'Donoghue, T., S. Scotchmer, and J.-F. Thisse. 1998. Patent Breadth, Patent Life, and the Pace of Technological Progress. *Journal of Economics and Management Strategy*. 7(1): 1-32.
- Olson, M. 1971. *Logic of Collective Action: Public Goods and the Theory of Groups*. Cambridge: Harvard University Press.
- Parker, G., M.W. Van Alstyne. 2005. Innovation through optimal licensing in free markets and free software. Available at. <http://ssrn.com/abstract=639165>.
- Patent Commons project. 2005. About Commitments. Available at [http://www.patent-commons.org/resources/about\\_commitments.php](http://www.patent-commons.org/resources/about_commitments.php)
- Paul, R. 2008a. Sun eclipses Red Hat Firestar pact as patent invalidated. Web log comment. Available at <http://arstechnica.com/open-source/news/2008/07/sun-eclipses-red-hat-firestar-pact-as-patent-invalidated.ars> (Accessed December 12, 2011).
- Paul, R. 2008b. Barracuda bites back at Trend Micro in ClamAV patent lawsuit. Web log comment. Available at <http://arstechnica.com/open-source/news/2008/07/barracuda-bites-back-at-trend-micro-in-clamav-patent-lawsuit.ars> (Accessed December 12, 2011).
- Png, I.P.L., Q. Wang. 2009. Copyright law and the supply of creative work: Evidence from the movies. Available at.  
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.151.4630&rep=rep1&type=pdf>.
- Prince, D.W., P.H. Rubin. 2002. The effects of product liability litigation on the value of firms. *American Law and Economics Review*. 4(1) 44.
- Rainer, A., S. Gale. 2005. Evaluating the quality and quantity of data on open source software projects. *Proceedings of the 1st Internat. Conf. Open Source Systems*, Genova, Italy.

- Romanosky, S., R. Telang, A. Acquisti. 2008. Do data breach disclosure laws reduce identity theft? Seventh Workshop on the Economics of Information Security. Center for Digital Strategies, Tuck School of Business, Dartmouth College, Hanover, NH.
- Ryan M. R., E. L. Deci. 2000. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary Educational Psychology*. 25(1): 54-67.
- Rysman, M., T. Simcoe. 2008. Patents and the Performance of Voluntary Standard Setting Organizations. *Management Science*. 54(11): 1920-1934.
- Roberts, J.A., I.H. Hann, S.A. Slaughter. 2006. Understanding the motivations, participation, and performance of open source software developers: A longitudinal study of the Apache projects. *Management Science*. 52(7) 984-999.
- Rooney, P. 2006. FireStar files suit against Red Hat. Computer Reseller News. Available at. <http://www.crn.com/news/applications-os/190300990/firestar-files-suit-against-red-hat.htm>.
- Samuelson, P., R. Davis, M.D. Kapur, J.H. Reichman. 1994. Manifesto concerning the legal protection of computer programs. *Columbia Law Review*. 94 2308-2431.
- Sanders, T. 2006. Red Hat slapped with patent suit. VNU Business Publications. Available at. [www.vnunet.com](http://www.vnunet.com).
- Schankerman, M. and S. Scotchmer. 2001. Damages and injunctions in protecting intellectual property. *RAND Journal of Economics*. 32(1): 199-220.
- Schumpeter, J. 1939. *Business Cycles*. New York: McGraw-Hill Book Company.
- SCO. 2003. Complaint in the Caldera Systems, Inc. (d/b/a The SCO Group) vs. International Business Machines Corporation case. Available at. <http://www.sco.com/scosource/complaint3.06.03.html>.
- Scotchmer, S. 1991. Standing on the shoulders of giants: cumulative research and the patent law. *The Journal of Economic Perspectives*. 5(1) 29-41.
- Scotchmer, S. 2005. *Innovation and Incentives*. MIT Press, Cambridge, MA.

- Seeker, S. 2010. The Defensive Patent License makes patents less evil for open source. NetworkWorld.
- Serafino, D. 2007. Survey of Patent Pools Demonstrates Variety of Purposes and Management Structures. KEI Research Note 6.
- Shapiro, C. 2001. Navigating the Patent Thicket: Cross Licenses, Patent Pools, and Standard-Setting. In A. Jaffe, J. Lerner, S. Stern, eds. *Innovation Policy and the Economy*, Vol. 1. MIT Press, Cambridge, MA.
- Silverman, B.S. 1999. Technological resources and the direction of corporate diversification: Toward an integration of the resource-based view and transaction cost economics. *Management Science*. 45(8): 1109-24.
- Singh, P.V. 2010. The small world effect: The influence of macro level properties of developer collaboration networks on open source project success. ACM Transactions on Software Engineering and Methodology.
- Singh, P.V., Y. Tan, V. Mookerjee. 2011. Network effects: the influence of structural social capital on open source project success. *Management Information Systems Quarterly*. 35(4): 813-829.
- Smith, M.D., R. Telang. 2009. Competing with free: The impact of movie broadcasts on DVD sales and Internet piracy. *Management Information Systems Quarterly*. 33(2) 312-338.
- SourceForge. 2009. What is SourceForge.net? Available at.  
<http://sourceforge.net/apps/trac/sourceforge/wiki/What%20is%20SourceForge.net>  
?
- Stevens, K. 2012. JBoss enterprise middleware. Red Hat - VDEL Workshop. Available at <http://www.vdel.com/files/redhat/JBossEnterpriseMiddlewareOverview.pdf>
- Stewart, K.J., A.P. Ammeter, L.M. Maruping. 2006. Impacts of license choice and organizational sponsorship on user interest and development activity in open source software projects. *Information Systems Research*. 17(2) 126.

- Stewart, K.J., S. Gosain. 2006. The impact of ideology on effectiveness in open source software development teams. *Management Information Systems Quarterly*. 30(2): 291-314
- Shah S.K.. 2006. Motivation, governance, and the viability of hybrid forms in open source software development. *Management Science*. 52(7): 1000-1014
- Stoneman, P. 2002. *The economics of technological diffusion*. Blackwell Publishers Ltd, Oxford.
- Subramaniam, C., R. Sen, M.L. Nelson. 2009. Determinants of open source software project success: A longitudinal study. *Decision Support Systems*. 46(2): 576-585
- Takahashi, D. 2003. SCO Group says it has revoked IBM's license for Unix. San Jose Mercury News California.
- Thompson N.C., D.C. Mowery, A.A. Ziedonis. 2011. Materials transfer agreements (MTAs), licenses, and the flow of scientific knowledge. NBER working paper
- Thursby, J., R. Jensen, M.C. Thursby. 2001. Objectives, characteristics, and outcomes of university licensing: a survey of major U.S. universities. *Journal of Technology Transfer*. 26(1-2): 59-72.
- Tucker, C. 2012. Patent Trolls and Technology Diffusion. Working Paper, Sloan School of Management, MIT.
- Van Antwerp, M., G. Madey. 2008. Advances in the SourceForge Research Data Archive (SRDA). The 4th International Conference on Open Source Systems, Milan, Italy.
- Viscusi, W.K., J. Hersch. 1990. The market response to product safety litigation. *Journal of Regulatory Economics*. 2(3) 215-230.
- von Hippel, E. 2006. *Democratizing innovation*. MIT Press, Cambridge.
- von Hippel, E., G. von Krogh. 2003. Open source software and the “private-collective innovation model: Issues for organization science.” *Organization Science*. 14(2): 209-223.

- Walsh, J.P., A. Arora, and W.M. Cohen. 2003. Working through the patent problem. *Science*. 299(5609): 1021.
- Walsh, J.P., W.M. Cohen, and C. Cho. 2007. Where excludability matters: Material versus intellectual property in academic biomedical research. *Research Policy*. 36(8): 1184-1203.
- West, J. 2003. How open is open enough? Melding proprietary and open source platform strategies. *Research Policy*. 32(7) 1259-1285.
- Windmeijer, F. 2000. Moment Conditions for Fixed Effects Count Data Models with Endogenous Regressors. *Economics Letters*. 68(1): 21-24.
- Windmeijer, F., J. Santos Silva. 1997. Endogeneity in Count Data Models: An Application to Demand for Health Care. *Journal of Applied Econometrics*. 12(3): 281-294.
- Wooldridge, J.M. 1997. Multiplicative Panel Data Models without the Strict Exogeneity Assumption. *Econometric Theory*. 13: 667-678.
- Wooldridge, J.M. 2002. *Econometric analysis of cross section and panel data*. Cambridge: The MIT press
- Ziedonis, R.H. (2004). Don't fence me in: Fragmented markets for technology and the patent acquisition strategies of firms. *Management Science*. 50(6): 804-820.