DOCTORATE IN CO-ASSOCIATION BETWEEN
TELECOM ECOLE DE MANAGEMENT & UNIVERSITY EVRY VAL D'ESSONNE

Speciality: Management Science
Doctoral School: Science of Society

Presented by

Mr. Altay ÖZAYGEN

To obtain the degree of

DOCTOR OF MANAGEMENT SCIENCE
OF
TELECOM ECOLE DE MANAGEMENT

ECONOMIES OF SCOPE, MODULARITY and PROPERTY RIGHTS in SOFTWARE TECHNOLOGIES

Defended on 6 December 2013

Director of thesis:	**Mrs. Madeleine Besson** Professor, HDR, Télécom Ecole de Management
Co-Director of thesis:	**Mrs. Müge Ozman** Associate Professor, HDR, Télécom Ecole de Management
Reporters:	**Mrs. Jackie Krafft** Professor, HDR, CNRS - Groupe de Recherche en Droit, Economie, Gestion (GREDEG) de Nice Sophia Antipolis **Mrs. Kristina B. Dahlin** Associate Professor, Strategy & Business Policy, HEC Paris
Examinators:	**Mr. Gérard Kœnig** Professor, HDR, IAE Gustave Eiffel, Ecole de Management, Université Paris-Est Créteil **Mr. Gregorio Robles** Associate Professor, Grupo de Sistemas y Comunicaciones (GSyC), Universidad Rey Juan Carlos **Mr. Nicolas Jullien** Associate Professor, Logique des usages, sciences sociales et de l'information (LUSSI), Télécom Bretagne

These n°: 2013EVRY0029

Acknowledgments

I feel indebted to Müge Özman, my PhD advisor, for her helpful and critical comments that she gave me throughout this nearly four years. While providing total liberty in my research, she intervened when I was lost and showed me how to find my way. Her understanding and patience during this period was invaluable. I also thank Madeleine Besson for her support throughout these years.

I also thank participants to the DIMETIC 2011 Spring and Fall, DRUID 2012, GREDEG 2012 and DRUID ACADEMY 2013 who helped me to shape this work with their comments.

I am grateful to Jackie Krafft, Arianna Martinelli, Bulat Sanditov and Grazia Cecere who gave me comments on various part of this thesis. I thank Alison Bunel for her corrections made on the Résumé.

The research leading to this work was carried out in the context of the joint research project MODEUS/UBIMEDIA, between Institut Télécom (France) and Alcatel-Lucent Technologies. I thank to Alcatel-Lucent Technologies for their financial support.

During this thesis I used free and open source software. Main tools that I have used during this thesis are R, Gephi, Pajek(shareware), Python, LaTeX, Emacs, all on a Debian GNU/Linux distribution. I am indebted to all researchers, developers who have created these wonderful tools and toys.

I would like to thank my brother Alkım with whom I discovered Internet, computers and free software back in 1993. Since then we continue to share technical tips, discuss the development of these technologies and how they are affecting our own life. These discussions are the seeds of this work. I thank Cédric Gossart who helped me with our lengthy conversations on how to conduct a research and how to do a PhD. His encouragements were helpful. I also thank my good old friends Hakan Ertürk and Emre Altuğ Yavuz who have finished their PhD years before and gave me support and friendship throughout this journey. I am thankful to my friends Özgür Tek who proof-read this manuscript and Egemen Metin Turan who gave me some tips on Python script development.

I thank Hande Gözükan for the support and love that she gave me during these PhD years. Her encouragements gave me the force to start and finish this PhD. Moreover, I am thankful for her patience in helping me to understand a little bit more test driven object-oriented programming. I should also thank my parents who gave me their moral support. Without them I would not have been able to be in research for so long years. I am also thankful to my little friend Léo who has supported me by proposing to write an article together just after he has learned to read and write by his own.

Abstract

This thesis studies the role of the intellectual property right (IPR) regimes and its implications on the software industry. Moreover, it focuses on the balance of proprietary and free, libre and open source software (FLOSS) development models for innovative performance. In this thesis, first the concept of key factor of the fifth technological revolution, age of information and telecommunication is discussed within the techno-economic paradigm literature (Freeman and Perez, 1988). Then the three aspects of software technologies are investigated. The first two are the structural characteristics of software which are; economies of scope (Panzar and Willig, 1981; Teece, 1980) and modularity (Parnas, 1972; Langlois and Robertson, 1992). The last one is the IPR regimes which create the institutional peculiarities of software (Mazzeloni and Nelson, 1998b). In this thesis, different methodologies are used and several software technologies are considered to illustrate the research questions. Economies of scope of software is investigated by using an agent based simulation. The research on modularity is carried out by exploiting a patent analysis on the video indexing technology. The IPR issue is examined within two different chapters. In the first one, panel data analysis is used to understand the effect of patenting and contribution to the Linux kernel project on the performance of firms. The second one focuses on a case where open innovation is implemented by a software R&D group within Alcatel- Lucent Bell Labs, Nozay, France. These unique examples cannot give any macro trend on software industry but this thesis aims to contribute to the IPR discussions within the software industry.

Key words: software industry, modularity, free software, open source, Linux kernel, patent analysis, open innovation.

Résumé

Cette thèse étudie le rôle des différents régimes de propriété intellectuelle (DPI) et évalue ses conséquences sur l'industrie du logiciel. En outre, ce travail porte sur l'équilibre entre deux modèles de logiciels, celui des logiciels privés et celui des logiciels libres et open source, et cherche à évaluer leurs effets sur la performance des entreprises. Ainsi nous discutons dans un premier article les facteurs clés de la cinquième révolution technologique à travers le concept de paradigme techno-économique (Freeman et Perez, 1988) et nous considérons l'*open source* comme le principal de ces facteurs. D'autre part, nous étudions les trois aspects des technologies logicielles. Les caractéristiques structurelles des logiciels, c'est-à-dire les économies de gamme (Panzar et Willig, 1981; Teece, 1980) et la modularité (Parnas, 1972; Langlois et Robertson, 1992) représentent les deux premiers. Le régime de propriété intellectuelle, qui est à l'origine des particularités institutionnelles du logiciel (Mazzeloni et Nelson, 1998b), représente le troisième aspect. Au sein de cette thèse nous utilisons différentes méthodologies et considérons plusieurs technologies logicielles pour répondre à nos questions de recherches. Les économies de gamme de logiciels sont étudiées à travers une simulation multi-agents. La recherche sur la modularité est effectuée par une analyse des brevets sur la technologie d'indexation de vidéo. La question des DPI est examinée dans deux chapitres différents. Dans un premier chapitre, une analyse de données de panel est faite pour démontrer l'effet du brevetage et de la contribution au projet du noyau Linux sur la performance des entreprises. Le second chapitre traite quant à lui d'un cas particulier où l'innovation ouverte est réalisée par un group de recherche en ingénierie du logiciel au sein d'Alcatel-Lucent Bell Labs, Nozay, France. Ces exemples uniques ne peuvent conclure sur aucune tendance macro sur l'industrie du logiciel, mais cette thèse vise à alimenter les discussions sur les droits de propriété intellectuelle au sein de l'industrie du logiciel.

Mots clés: industrie de logiciel, modularité, logiciel libre, open source, noyau Linux, analyse de brevets, innovation ouverte, droits de propriété intellectuelle.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Information and communication technologies (ICTs) have an important impact on economy. Although the share of ICTs in the total production and employment is low, their rate of growth is high and by the end of the twentieth century, half of the US economic growth rate is credited to these new technologies (Freeman, 2007). ICT industries comprise computers, software, electronic components and telecommunication equipments. Among these, software technologies present an important peculiarity; information is the main economic input. Information has two important aspects; it is inexpensively reproducible and it is non-rivalrous. It is inexpensively reproducible because after the first copy, the bits that constitute the information can be transferred easily to many devices connected to the Internet. Its non-rivalrousness is due to its use in different sectors without any diminution in its content (Steinmueller, 2007).

Software technologies are cumulative and consist of various components provided by different sources. Software development is based on combination and recombination of inputs (Hall and MacGarvie, 2010). In this thesis, software technologies are investigated from three different perspectives. The first two are the structural characteristics of software technologies, which are *economies of scope* and *modularity*. The third is an institutional aspect which treats the *intellectual property rights (IPR) regimes.* Although many researchers have paid attention to the economies of scope (Steinmueller, 2007), to the modularity (Garud and Kumaraswamy, 1995; Langlois and Robertson, 1992; Sanchez and Mahoney, 1996; Baldwin and Clark, 1997; Fleming and Sorenson, 2001) and to the IPR related aspects of software technologies (Fosfuri et al., 2008; Bonaccorsi et al., 2006; Harison and Koski, 2010; Mazzoleni and Nelson, 1998b; Chesbrough, 2003a) separately, little is known about their interactions, and their influence on the innovation process and software production. Moreover, it is expressed that there is a need to find a balance between strong IPR regimes and open software development models for a sustainable innovation activities within the software industry (Lippoldt and Stryszowski, 2009). This thesis aims to contribute to the literature by filling these gaps.

The main research questions driving this work are as follows;

- What are the managerial implications to promote innovation in software sector as far as the IPR regime is concerned?
- How should firms balance free, libre and open source software (FLOSS) and proprietary development models for innovative performance?

To answer to these questions, various methodologies for different aspects of software technologies are used. Agent based simulation method is used to model economies of scope of the software industry. Patent analysis is carried out to understand the effect of modularity on the knowledge boundaries of firms contributions to the video indexing technology. Regression analysis and case study methodologies are used to discern the effects of IPR on firm outputs and innovation strategies.

In this chapter, the first section will present a historical perspective on the development of the software industry. Afterwards, the following section will give an overview of the concepts which are used in this thesis.

1.1 A brief history of the software industry

Theoretical works of Alan Turing, specifically his essay written in 1936 sets the foundations for software technologies although the term software was not used in this work. It is accepted that the first general purpose electronic computer which could be reprogrammed was ENIAC. It was constructed for the US Army during the Second World War for the calculation of the ballistic tables and other computations related to the development of the hydrogen bomb. Its presence was announced in 1946 after the war (Campbell-Kelly and Aspray, 1996). During the same period, similar efforts were made in Germany by Konrad Zuse and also at Harvard University. In 1941, Zuse commercialized the computer that he had developed for the Nazi regime. Nevertheless, due to the World War II, his inventions were unnoticed in the US (Ceruzzi, 2003). Moreover, both Zuse's computer and the project at Harvard were based on electromechanical computers. However, ENIAC was made of vacuum tubes.

The first spin-off in computer industry happened just after the World War II. Two lead engineers who worked in the ENIAC project had a dispute over patent rights and left the university to found their own firm. This company, which was later acquired by Remington Rand, built the UNIVAC mainframes. The first UNIVAC became well-known by predicting the results of the 1952 US elections. Firms like IBM, Burroughs and NCR also invested in this new business. In the late 1940s and early 1950s some thirty computer firms were established in the US. In 1951, Britain had its first computer on the market. However, the British computer industry had many difficulties due to business circles being reluctant to use computers (Campbell-Kelly and Aspray, 1996). It is acknowledged that the commercial computing started in the early 1950s (Ceruzzi, 2003). During this period, IBM developed the FORTRAN programming language for scientific calculations. It is still one of the dominant programming languages in many engineering fields requiring intense computing.

During the 1960s two important products were released. The first one is the System/360 mainframes from IBM and the second one is the PDP series of minicomputers made by DEC. These two products represent two different approaches to computing of the 1960s. Until the introduction of the Sytem/360 into the market, each computer system was using a unique software. System/360 product family introduced different computer hardware with varying performances but the same software were installed in each model. This new approach aimed to ease the upgrade process for computer users by eliminating the need to acquire a new skill set for each system. Moreover, it was possible to expand the host computer capabilities with over 40 different peripherals such as printers, storage and so on. At the end of the 1960s, IBM had three quarters of the market share in mainframe computers. System/360 became an industry standard and created software lock-in (Arthur, 1989) for IBM as well as for its customers. Due to compatibility issues users faced problems and difficulties in changing their systems to the systems of the competitors of IBM. For the same reasons IBM itself also remained closely attached to its 360 architecture (Campbell-Kelly and Aspray, 1996).

On the other hand, DEC introduced PDP series microcomputers which were more affordable than the IBM System/360 line of computers. The first UNIX operating system was written on a PDP microcomputer by AT&T researchers at Bell Labs. Among many novelties, UNIX system was designed to run on different hardware platforms. Moreover, the license fee of its source code was affordable for research units and for universities. The case of UNIX, releasing its source code for educative purposes, further debugging and refining, is an early cultural and technical precursor of the free software movement (Campbell-Kelly and Aspray, 1996; Ceruzzi, 2003; Weber, 2004).

In the beginning of the 1960s, software business started to flourish. In 1965, there were around 40 to 50 major independent software and programming service suppliers with several hundred small firms active in the US market (Steinmueller, 1995). In 1969, IBM decided to unbundle their software and System/360 computer product which was sold in a single package. Moreover, large companies and government needs created an important number of independent, small contract programming companies. In addition, between 1969 and 1971, the development of the microprocessor by Intel has dramatically changed the computer industry. Microprocessors helped the production of low-cost computers enhancing the diffusion of computers. In January 1975, the first microprocessor-based computer Altair 8800, which was the first PC for an electronic hobbyist, was released. These important changes in computer industry helped the creation of new software companies developing packaged software (Steinmueller, 1995).

After 1975, few computer programs targeting the consumer market showed the importance of the software industry. The BASIC programming language interpreter for Altair 8800, released in 1975, was the first product of Microsoft then called Micro-Soft. Altair computer was targeting hobbyist niche market where people had the culture of sharing both their experiences and their programs. Those people also shared the BASIC interpreter among them. As a reaction, Bill Gates wrote an open letter to hobbyists to stop sharing commercial software products. It was the start of a new era in which software was closed source and sharing was a violation of the licensing agreements.

By the end of the 1970s, two important software products were released. In 1979 VisiCalc, the first spreadsheet software was launched. Between 1978 and 1979, full WYSIWYG (what you see is what you get) word processor WordStar gained two-third of market share. By 1980, there were nearly a dozen spreadsheet and word processing software packages on the market (Campbell-Kelly and Aspray, 1996). These packaged software helped the acceptance of the computer technology by the masses.

During 1982-83, IBM PC became an industry standard. Companies which resisted to the standards imposed by IBM went out of business. Only Apple out of that era has competed with IBM PC and still withstands. According to Campbell-Kelly and Aspray (1996) Apple resisted to the market pressure imposed by IBM PC by producing better software.

Concurrently, during the 1980s a new software development and distribution culture emerged around the Free Software Foundation (FSF). FSF started the GNU project which aimed to provide a complete operating system that is solely free software. FSF played a significant role on the history of software with the software license, General Public License (GPL), that it has created. GPL gives the users the right to obtain and study the source code, and the liberty to modify it and to distribute it as long as the distribution includes the source code and the GPL text (Stallman, 2002; Weber, 2004). GPL aims to protect software users and sustain the development of the code by forcing it to be accessible for its users for any purpose. Since its foundation in 1984, FSF promotes the idea of free software and continues to develop GNU software. The importance of the FSF is its manipulation of the intellectual property rights regime to disseminate and sustain the development of software.

The idea of sharing technical knowledge within the software industry did not start with the free and open source software movement. Historically, sharing, co-creation and dissemination of new ideas has been a rule in scientific and artistic environments (Graham, 2004). One of the first historical examples of co-creation within the software industry was the collaboration of engineers working in different companies for the development of a compiler targeting the IBM 705 hardware in 1952 (Weber, 2004). Another example is the reference manual of the IBM 704 SHARE library which promoted sharing as a rule of conduct in 1959 (Nash, 2010). In 1961, DEC Users' Society; DECUS was founded. Members of the DECUS were programmers working with DEC computers. Membership to DECUS was not allowed to DEC engineers. The aim of the DECUS was to share technical knowledge among DEC users. DEC relied on DECUS to get in touch with customers to obtain proper feedback about their products (West and Gallagher, 2006).

After the mid 1980s, one of the interesting examples in computer industry was Sun Microsystems. The company adopted open standards and based its operating system on an open source project, Berkeley Software Distribution (BSD), developed at Berkeley University. Sun Microsystems produced workstations which used SPARC, designed by Sun, as well as Intel based processors (Weber, 2004). To expand its market share, Sun Microsystems facilitated the access of other firms to its technologies by adopting open standards (Garud and Kumaraswamy, 1993). With the acceptance of FLOSS and its development model, in 1998, Open Source Initiative (OSI) was founded. It aimed to push FLOSS development model and its usage within the business circles with more pragmatic motivations.

During the 1990s it was clear that the software industry gained an important part of the gross domestic product of developed countries (Lippoldt and Stryszowski, 2009). Dot-com bubble spanning from 1997 to 2000 showed the importance of the software industry not only for US but also for many other

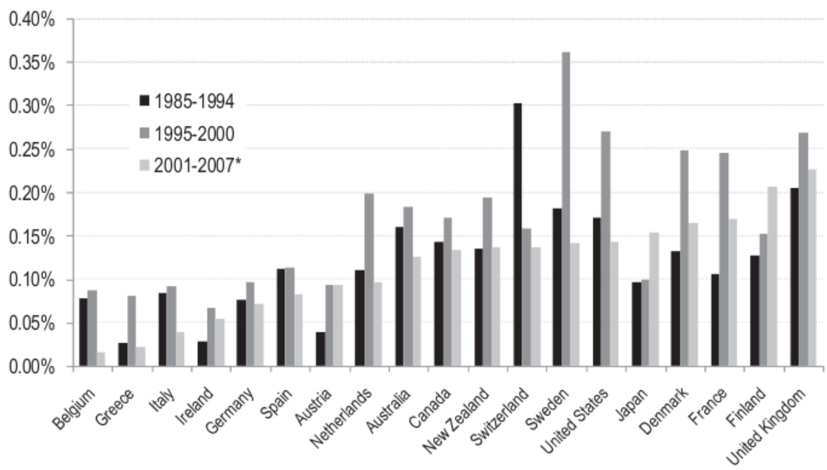


Figure 1.1: Contribution of software investment to GDP growth (Lippoldt and Stryszowski, 2009).

countries. Software industry, with its easy entry level, played an important role in the economic development of some emerging countries (Arora and Gambardella, 2006). Today, software industry has a substantial effect on the economic growth of many countries' as given in Figure 1.1.

1.2 Themes treated in this thesis

This thesis exploits some well-defined concepts from the field of innovation studies and management to investigate the relationship of structural and institutional characteristics of software technologies in relation to IPR and innovative activities within the software industry. This section gives an overview of these concepts and how they are applied in the thesis.

Key factor is a concept developed by Freeman and Perez (1988) within the techno-economic paradigm literature which is a follow-up work of the long-wave theory (Freeman and Louça, 2002). This theory analyses the long term fluctuation of capitalism. The techno-economic paradigm has a systemic view of the economy with a focus on technology, organization and culture, all interacting with each other. Each techno-economic paradigm relies on some raw material as an input required by the technology. Key factors have an important influence on the new social structure as well as on the new techno-economic paradigm. Key factors should have (1) relatively low and rapidly falling costs, (2) unlimited supply over a long period of time and (3) possibilities to be used in a large number of products and process through out the economy. In Chapter 2, it is argued that FLOSS complies with the definition of the key factor and it is claimed that FLOSS is the key factor of the fifth techno-economic paradigm.

Economies of scope describe cost saving through the combination of two or more production lines in one firm rather than conducting them separately (Panzar and Willig, 1981). According to Steinmueller (2007), economies of scope in the production of ICTs are based on the deployment of similar designs which satisfy the requirements of different applications. This is achieved by maximizing the reuse of earlier design models, creating important possibilities for product differentiation in ICTs.

In Chapter 3, economies of scope in software industry are investigated through an agent based simulation model on alliance formation between firms. The literature on inter-firm alliances gives an emphasis on the technical competencies of partners and posits that the complementarities between firm resources are a major motivation to form alliances (Hagedoorn, 1993; Shan et al., 1994; Eisenhardt and Schoonhoven, 1996). Within the evolutionary framework, although various approaches exist to measure complementarities, usually the technological distance (or overlap) between firms is used (Mowery et al., 1998; Rothaermel and Boeker, 2008). However, there are many examples within the software industry which show that alliances are not only created depending on the technological distance but also depending on the market distance to get access into different markets. As an example, a product running on an embedded software, can be adapted with small changes in the software and if required on the hardware in order to be introduced into different markets. However, if the producer does not have enough resources to make the changes to the product, it can search and explore a partner which would complement the missing capabilities. The missing parts are either technological or market related or both. Through an agent based simulation study, the learning dynamics and evolving networks are investigated when firms form partnerships in a regime rich in economies of scope.

Modularity is the third concept explored in relation to the software industry. Research on software modularity dates back to 1970s, to the early stage of the software engineering. The theory of modularity in software engineering is set by Parnas (1972). Alongside the modularity in software, he also sets the principles of object oriented programming. Software modularity aims to develop reliable software in a short period of time through reuse of previously written code. Various scholars have explored modularity in different technologies (Garud and Kumaraswamy, 1995; Langlois and Robertson, 1992; Sanchez and Mahoney, 1996; Baldwin and Clark, 1997; Fleming and Sorenson, 2001). The literature on modularity argues that modularity increases specialization at the firm level. However, there is not much research conducted on the effects of modularity on software industry. Chapter 4 and 5 aim to fill this gap.

The research questions of these chapters are; 1) Do technological achievements in a highly modular software technology originate from firms which are diverse or which are specialized? 2) What are the knowledge boundaries of firms contributing to the development of a software technology? As a methodology, patent connectivity analysis (Verspagen, 2007) is used to determine firms which have made technological achievements. Video indexing, which classifies and retrieves information automatically from video content, is studied as a case. Modules of the video indexing technology are also investigated by using patent connectivity analysis. These are optical character recognition (OCR), audio and speech analysis, and image analysis. In this chapter, it is found that the firms which have determined the evolution of video indexing and its modules have high technological diversity.

IPR regimes are an institutional feature of software technologies. IPR has an important place in the development of the software industry. As mentioned previously, the first spin-off within the computer industry was due to a patent dispute in 1946. There are two opposite approaches to develop business models and to create technology within the software industry. These opposite approaches are in close connection with different IPR regimes. On one side, firms extensively use IPR strategies such as patenting and licensing to monetize the users. On the other side, some of these firms use collaboration to develop some of their software which are distributed through the Internet with a free access. "Open" approaches to do business and achieve technological development became important with the proliferation of FLOSS development and the development of the related business models. In Chapter 6 and 7 the effect of IPR on software industry is inquired.

Chapter 6 aims to contribute to the literature by examining the effects of patenting and FLOSS development activities on the performance of a firm through econometric evidences. This empirical work is limited to the development of the Linux kernel project which is one of the most successful flagships of the FLOSS movement. This study collects data on 169 firms among 800 firms which have contributed to the Linux kernel during seven years (2005-2011). During this period, these 169 firms have contributed to the 48.7 % of the total changeset (a group of modification containing files which are relevant to each other) accepted by the project. The research question which guided this chapter is: How does the contribution to Linux kernel project and patenting affect the firm performance and are these two activities complementary? In this research, it is found that contributing to the Linux kernel project and patenting positively affect sales of firms, however they are substitutes. In other words, there is a negative interaction effect between them as far as they affect sales.

Chapter 7 presents a case study on the difficulties of putting "open innovation" concept in practice by a software R&D group in Alcatel-Lucent Bell Labs, Nozay, France. There are many company case-studies that highlight successful implementation of open approaches for development and commercialization of technology under the term "open innovation" where "open innovation" is defined as "the use of purposive inflows and outflows of knowledge to accelerate internal innovation, and expand the markets for external use of innovation, respectively" (Chesbrough, 2006). This chapter aims to identify barriers of adopting open innovation within an ICT firm. This process is discussed within the concepts of routines (Nelson and Winter, 1982) and "interpretive schemes" i.e. the difference in inter-organizational thinking and collaboration styles among different departments in large firms (Dougherty, 1992).

1.3 Methodologies used in this thesis

In this thesis various methodologies are used to answer the above research questions. The use of different methods are not related to random preferences. One of the main reason is related to the scale of researched unit. Moreover, the decision of using a particular methodology also lies on the availability of required data and time constraint. However, without doubt, there is also the will of experimenting and learning different research methodologies.

Chapter 2 is a conceptual paper which argues that FLOSS is a key factor of the fifth Kondratiev wave. This chapter is based on the techno-economic paradigm literature Freeman and Perez (1988). The definition given for the concept of the key factor is the same authors. This chapter is a conceptual paper because a new key factor is proposed and discussed based on its definition.

The methodology used in Chapter 3 is agent based simulation. In the literature related to the agent based simulations various advantages are given. According to Axelrod (2003) agent based simulations can address important problems and can help to predict with complicated inputs with hypothesized mechanism. Simulation can also help to validate and improve the model on which a simulation is based. Axelrod (2003) emphasize that in social sciences even highly complicated models are rarely proven to be accurate but social scientist discovered important relationships and principles by using simulations which are based on very simple models. One of the classic examples in this area is developed by Schelling (1978). According to Gilbert and Terna (2000), in addition to argumentation and formalization, agent based simulations is the third way of carrying out social science.

Arthur (1994) argues that human behavior is based on "localized deductions based on our current hypothesis" which is an inductive reasoning. He has illustrated how to model inductive reasoning with the now famous El Farol Bar problem. Agent based simulation allows to formalize complex theories, carry out experiments and observe emergence within a system. Agent based models frees the researcher from the consideration of provability and it allows to focus on the most relevant aspect of the world that the researcher believes (Page, 2005). Moreover, according to the author, the researcher is constrained with his/her imagination and interest.

In Chapter 3, the aim is to understand how learning level of firms changes according to different IPR regimes and knowledge codifications. For such a complex problem an agent based simulation is conducted by using few parameters i.e. modeling is carried out with a KISS approach (Fagiolo et al., 2007).

In Chapter 4 and 5 with respect to the research questions it is crucial to find out firms which have made considerable technological achievements in a certain technological field. For this purpose a patent analysis technique, which was first developed by Hummon and Doreian (1989) to find out important scientific paper by using their citation connection networks which is also then refined for patents citation networks by Verspagen (2007), is used. From the information obtained with this methodology, knowledge diversity of firms based on their patent portfolio is conducted.

Patent analysis is an important tool to understand the knowledge of firms. The drawback of this tool is that not all inventions are recorded with patents and some firms could even avoid to patent by adopting a trade secret strategy. In Chapter 6 another patent counting is carried out for an econometric analysis.

In Chapter 6 the research question is related to firms which are patenting but also at the same time giving away for free software developed in-house. The motivation behind this chapter is to find out how these two IPR strategies affect the performance of firms. To answer the research questions of this chapter the FLOSS Linux kernel project contributing firms are investigated. Contributing firm names are obtained from the project web site, patenting activities are measured by patent counting and firm performances are obtained from a specialized financial database. The aggregate nature of data make it obvious to conduct an econometric analysis in this chapter.

In Chapter 7, the unit analysis is a software R&D group and the research question is related to the obstacles encountered by the R&D group in their adoption of the open innovation strategy. The smallness of the unit of analysis and data which could be obtained to answer the research questions impose the use of case study. This methodology gives enough room to uncover various actors and reasons which represent important part in the barriers of open innovation strategy adoption that no other methodology could uncover.

Table 1.1: Themes treated in this thesis.

Chapter	Theme	Method	Research Questions	Conclusion
Ch.2	Techno-economic paradigm	Conceptual paper		FLOSS is the key factor of the fifth techno-economic paradigm
Ch.3	Economies of scope	Simulation	1) What are the dynamics of learning in a system with rich economies of scope? 2) To what extent does learning depend on IPR regime and knowledge codification?	Firms with alliance preferences close in one dimension, either market or technology, and far in the other dimension are better-off.
Ch.4-5	Modularity	Patent analysis	1) Do technological achievements in a highly modular software technology originate from firms which are diverse or which are specialized? 2) What are the average knowledge boundaries of firms contributing to the development of a software technology?	Firms having an impact on the development of the technology consist of mainly highly diverse, multi-technology firms and a small number of specialized firms.
Ch.6	IPR	Panel data	How does the contribution to the Linux kernel project and patenting contribute to the performance of firms and are these two activities complementary?	Patenting and contribution to the Linux kernel project are positively correlated with sales. However, their interaction is negative i.e. they are substitute.
Ch.7	IPR	Case study	What are the difficulties in putting "open innovation" into practice for a software R&D group within a multinational ICT company?	The firms should distinguish the difference between hardware and software IPR management for a balanced approach towards software IPR.

Chapter 2

Free, libre and open source software as a key factor

In this chapter the concept of *key factor* will be reviewed in relation to FLOSS. The concept of key factor has been built within the techno-economic paradigm literature developed by Freeman and Perez (1988). Techno-economic paradigm is a follow-up work of the long-wave theory which is accepted as being developed by Kondratiev in 1920s. The long-wave theory aims to explain long term fluctuations of capitalism. Kondratiev uses both early econometric techniques and the historical account as a method to analyze these fluctuations. Schumpeter names long waves as "Kondratiev cycles" to honor the Russian economist (Freeman and Louça, 2002).

Each technological revolution brings a "transformation of the institutions of governance, of society and even of ideologies and culture [...] leading to intense and sometimes violent social tensions" (Perez, 2002). The social and institutional friction, which occurs in the information techno-economic paradigm, is between the technical advances facilitating distributed bottom-up organizations and the extension of the intellectual property rights. This friction also has various political resonance. In various countries Pirate parties, which have among others, the aim to abolish the copyright and the patent system, are founded.

Next in this chapter, the concept of key factor will be given then the case of FLOSS as a key factor is presented and the final section concludes.

2.1 Key factor

Long-wave cycles start with the industrial revolution and generally these waves span from forty to sixty years. They are caused by the emergence of certain key technologies and industries adopting a new key technology are attracted by the cost advantages (Perez, 1983). The diffusion of the technical change creates social and institutional frictions causing a structural crisis of adjustment until a better match is reached. Once this match is achieved a long term, stable investment atmosphere starts lasting two or three decades (Freeman and Perez, 1988). According to Perez (1983) long-waves are very complex and society-wide processes.

According to this theory, cycles are composed of alternating periods of high and relatively slow economic growth (Freeman and Louça, 2002). In one long-wave cycle there are two different periods which are distinguished by their patterns of innovation. These two periods are known as Schumpeterian pattern of innovation of the Mark I and Mark II type (Nelson and Winter, 1982). In Mark I pattern of innovation, there are ample technological opportunities creating high rates of entry, low appropriation of innovation and low cumulativeness of technological advances at the firm level and there is a limited role of generic knowledge. On the other hand, in Mark II pattern of innovation, there are less technological opportunities with low rates of entry, high appropriation of innovation, high cumulativeness of technological advances and a knowledge base closer to basic science (Breschi et al., 2000). Sidney Winter names these two phases as "entrepreneurial" (Mark I) and "routinized" (Mark II) modes of innovation, these two phases are in short grounded on knowledge base, entry conditions and selection (Fagerberg, 2003).

There are two different research strands aiming to verify Mark I and Mark II theories. The first is the older tradition which centered their work on the properties of firms such as its size, monopolistic power etc. The second one attempted to explain these theories according to the industry life cycle view

(Malerba, 1996). According to the industrial life-cycle view, technology is changing rapidly, entry barriers are low and new firms are conducting the large proportion of innovative activities creating a positive business environment during the first stage of an industry. However, during the later stage, technology matures with a concrete trajectory, entry barriers increases for new firms and financial resources become important in the competition between firms (Klepper, 1997). Thus an indication and confirmation that Mark I and Mark II phases are related to the technological class and technological regimes (Malerba, 1996; Breschi et al., 2000).

The techno-economic paradigm which is developed upon the long-wave theory has a systemic view of the economy which encompasses technology, organization and culture, all influencing each other. The start of a new techno-economic paradigm is recognized by clusters of technical, organizational and managerial innovations which are found in several industries with a pervasive effect on the whole economy. This trend continues until the technology generates profit while being refined. It is also observed that the new paradigm has an important influence on other aspects of the society (Freeman and Perez, 1988).

Freeman and Perez (1988) define the key factor from an economic point of view. Key factors are changing the technological systems and thus creating a new social structure as well as the new techno-economic paradigm. Each of the techno-economic paradigm relies on raw material as an input required by the technology. Key factors should have (1) relatively low and rapidly falling costs, (2) unlimited supply over a long period of time and (3) possibilities to be used in a large number of products and process through out the economy. Table 2.1 portrays the five techno-economic paradigms along with its related important technology, the year of its launch and the key factor.

Key factors usually exist even before a new techno-economic paradigm develops. However, new key factors are recognized after the previous techno-economic paradigm and related technologies signal diminishing returns and showing some limits in their refinements. The new key factor appears at the core of the rapidly growing system of technical, social and managerial innovations and it is not an isolated input. The new key factor triggers successive innovations by means of a feed-back mechanism and by showing the limits of the old paradigm. The new technology system arises as the new, ideal type of production organization. This new production organization becomes the common sense of management and design by attracting managers to the new techno-economic paradigm rules. Furthermore, the new key factor creates widening investment opportunities by signaling substantial increases in productivity and profits (Freeman and Perez, 1988).

Table 2.1: Industrial revolutions with key factors (Perez, 2002).

Technological revolution	Popular name	Big-bang technology	Year	Key factor
FIRST	The 'Industrial Revolution'	Arkwright's mill	1771	Cotton and iron
SECOND	Age of Steam and Railways	Steam engine between Manchester Liverpool	1829	Coal and transport
THIRD	Age of Steel, Electricity and Heavy Engineering	Bessemer steel plant	1875	Steel
FORTH	Age of Oil, the Automobile and Mass Production	Ford Model-T	1908	Oil
FIFTH	Age of Information and Telecommunications	Intel microprocessor	1971	Microprocessor

2.2 The FLOSS case

The present wave is accepted as the information technology paradigm and according to Goransson and Soderberg (2005), its key factor is *intuitively* indicated as the microprocessor. The key factor of the present wave, Intel's first microprocessor in 1971, is seen as the birth of the Information Age (Perez, 2002).

Goransson and Soderberg (2005) are suggesting that the key factor could be defined as the input among all other inputs in a technological system that most apparently lacks substitute and question whether it is the technological principle that should be regarded as the key factor in technological system rather than the physical inputs. They are also arguing that in the new information society there is a transition from energy, material and transport dominated economy into a high knowledge-content. In this chapter it is argued that many properties of FLOSS fit into the definition of the key factor.

A legitimate question would be why *proprietary software* is not also a key factor of the fifth techno-economic paradigm. Proprietary software is licensed with various restrictions to the users and generally the source code is not provided. The cost of production of proprietary software does not decrease. Instead while the development tools get better and more sophisticated, software users are also asking for more complex products properties. This trend requires highly trained human capital for the software production. A proprietary software which is in its end-of-life has very small chance to be released as open source. Even if this happens, a bulk of source code would have difficulties to attract new developers who will first understand the architecture and the code, then add his/her contribution to the software project. Thus proprietary software does not have an unlimited availability of supply. Due to the close structure of the code and restrictive licensing, a proprietary software does not have much opportunity to be modified and deployed in many of the products and process through out the economy. As a conclusion; proprietary software cannot be defined as a key factor due to its technical properties and the IPR regimes.

Low and rapidly falling relative cost

The cost of ownership of free and open source software is negligible. On the other hand, total cost of ownership is not null, training and migration to FLOSS have some costs but on the long run it is believed that with trained technical staff, FLOSS system administration is easier, safer and on the long run it costs less than other proprietary solutions (Varian and Shapiro, 2003).

The total cost would not be for the software itself but for the work force which would custom tailor the software to the requirements of the hardware and the users. Hence, this cost is very negligible in many cases compared to an effort to develop new software from ground up. It is estimated that it would take $1.4 billion to develop the GNU/Linux kernel alone and 10.8 billion to develop one of the popular GNU/Linux distribution which are all available on the Internet for zero cost and the source code is included (McPherson et al., 2008).

Unlimited availability of supply over long period

The cost of the software replication for FLOSS is negligible for those having an Internet connection. The General Public License protects the users to always have access to the source code in case they want to perform further developments. Moreover, back-compatibility of the code allows most of the time to run the code on different, old, low-cost hardware. As the code is open and protected by the GPL, forking i.e. creating a new project based on another FLOSS code base, retrieving a part of any project or using any discontinued project is possible. GPL allows the code to evolve as long as there is an interest from developers with different motivations (Dalle et al., 2008). This eco-system enables the unlimited supply of code for the future.

Clear potential for the use in many products and processes through out the economy

The new techno-economic paradigm shapes the productive organization and also the investment patterns (Perez, 1983). FLOSS development practices started to be used in other aspects of life which became a successful way to develop, do and manage business. Moreover, the "open source way of working", i.e. community based open source software development, has a potential to be a "paradigm-shifting" phenomenon, with consequences that would affect more than the software industry (David and Shapiro, 2008).

As the open source movement itself grows and evolves, the open source process is transposed to other industries (Lerner and Tirole, 2002). The success of the FLOSS development model is also reflected in more "open" approaches in R&D and product development which have been conducted in closed innovation environments for many years. FLOSS development methodologies and business models in relation to it are an influential example to the open innovation practices (Chesbrough et al., 2006, p.1). Open innovation literature is referred in Chapter 7.

The pervasiveness of FLOSS is difficult to show in numbers. FLOSS usage numbers as well as FLOSS OS (Linux and different BSDs) installations cannot be known accurately. There are various reasons behind this measurement problem. One of the important reasons is that FLOSS is free to copy and distribute and many FLOSS developers are encouraging others to distribute their software in order to minimize their own bandwidth consumption. For these reasons, users are motivated to distribute their copies and there are numerous distribution channels such as P2P, bit-torrent, FTP sites which makes it very difficult to make an assumption on the number of FLOSS users. Moreover, FLOSS is not only used on PCs, smart-phones etc. but also on numerous electronic appliances. Netcraft[1] statistics which are the most cited statistics on web servers have generally shown over 60% usage of FLOSS OS and FLOSS web server among all web servers on the Internet (Bonaccorsi and Rossi, 2003; Lerner and Tirole, 2002; Raymond, 1999). Even if any reliable statistics on this issue could not be shown it is proposed that nearly all Internet users are in some way connected to various FLOSS OS through sites such as Google, Facebook and many others. Moreover, with the increased usage of smart-phones and any other embedded devices in our daily life there is an important chance of using a FLOSS OS.

2.3 Discussion

Networking and distributed production are shown as one of the aspects of the information technology paradigm. Availability of the source code and its protection by the GPL or other similar licenses, new business practices such as open innovation and the mash-up culture, which combines pre-existing

[1] http://news.netcraft.com/archives/web_server_su rvey.html

text, video, music or programs aiming to create new ones, gains developer and consumer base among younger generations. These developments could hinder the domination of few big companies carrying out many innovative activities. Other than business practices such as open innovation, especially the free software and the GPL allow the innovations to be accessible for anyone. This property enables the development of new entries using FLOSS. Even incumbent firms of the information techno-economic paradigm such as Google, IBM etc. are contributing to FLOSS and some of their innovations are carried out in an open environment which could be used by other competing firms or by new entrants.

The significance of the long wave approach could be questioned in the Age of the Internet. Moreover, the definition of key factor has some similarities with the concept of *'General Purpose Technologies'* (GPT) (Bresnahan and Trajtenberg, 1995) which is generally treated with respect to the process of economic growth. GPT have tree characteristics: 1) Pervasiveness; the GPT should be used in most sectors, 2) inherent potential for technical improvements; the GPT should have a decreasing cost and increasing performances over time, and 3) innovational complementarities; the GPT should help to invent and develop new process and products Jovanovic and Rousseau (2005). The literature on GPT emphasis on the effect of the GPT on economic growth. Moreover, the development and the early phase of the technological development is not discussed within the GPT literature. However, being attached to the Schumpeterian analysis, the techno-economic paradigm literature emphasis on the early phase then the development of the key factor within a cyclic analysis of technological change. Thus, in this chapter FLOSS is analyzed within the techno-economic paradigm literature and argued that it is the key factor of the fifth long-wave.

After the fifth wave there is still not any consensus among scholars on the next wave. Various researchers also propose their own interpretation of technological cycles. In line with the techno-economic paradigm literature the next wave is proposed as the microelectronics-biotechnology era by Goransson and Soderberg (2005). But, as it is argued above, FLOSS will affect the next waves.

2.4 Conclusion

In this chapter, it is argued that FLOSS is one of the key factors of the information techno-economic paradigm. FLOSS fulfills the requirement of the key factors which are (1) relatively low and rapidly falling costs, (2) unlimited availability of supply over a long period of time and (3) clear potential for the use or incorporation of the new key factor(s) in a large number of products and process throughout the economic system either directly or through a set of related innovations. Moreover, it is also claimed that the social and institutional friction which occurs in the information techno-economic paradigm is the result of technical advances facilitating distributed, bottom-up organizations and the discussions on the intellectual property rights. Accordingly the main threat to the FLOSS comes from software patents and proprietary standards.

Chapter 3

Learning in regimes with rich economies of scope[1]

The ICT industries are different from other industries by the number of technological opportunities that could be exploited. In the production of ICTs, economies of scope are achieved by using similar designs for different application needs. This is realized by reusing of what has been learned from the design of earlier models (Steinmueller, 2007). There are many possibilities for product diversification through the application of software technologies into different markets which create rich economies of scope (Panzar and Willig, 1981). As an example FLOSS operating systems such as GNU/Linux or BSDs are used in various IT instruments like routers, servers, firewalls, etc., but also in products like refrigerators, watches, smartphones and wearable computers and so on. Nevertheless, these examples are not limited to FLOSS and there are also various examples from proprietary software products. Alliances are the means through which firms can diversify and create products for different markets. If capabilities of a software firm are not sufficient to diversify its products, that firm could search for partners which would complement its capabilities. Moreover, firms which have limited capabilities in software development also search for partners having more competencies in software technologies for diversification and improvement of services and

[1]Previous versions of this chapter are Ozman (2010) and Ozman and Özaygen (2011).

products.

The number of alliances, made by software firms in US, during 1990-2001, had a steady increase approaching to 95% of firms in 2001 (Lavie, 2007). A similar growth is observed for the period of 1970-1999 within the global software industry R&D partnerships (Cloodt et al., 2010). The literature on inter-firm alliances argues that complementarities between firm resources are a major motivation to form alliances and there is an emphasis on the technical competencies of partners (Hagedoorn, 1993; Shan et al., 1994; Eisenhardt and Schoonhoven, 1996). Lavie (2007) argues that alliances made with partners having abundant marketing and financial resources contribute to the market performance of software firms. Moreover, the author adds that alliances based on technology and human network resources are not sufficient to create any value.

This chapter focuses on learning from alliances by taking into account market and technical complementarities in the software industry. The research questions of this chapter are; 1) what are the dynamics of learning in a system with rich economies of scope? 2) To what extent does learning depend on IPR regime and knowledge codification?

In this chapter, the formation of strategic alliances and their results on learning of firms are analyzed. Firms form alliances with respect to knowledge and market complementarities in firm resources. Distance between firms are measured with respect to the similarities or disimilarities between firms. Knowledge distance between firms refers to the overlap between the technical competences of firms. Market distance, on the other hand, measures the overlap in their market domains. The model used in this chapter is inspired from the Saviotti and Metcalfe (1984) model of innovation which perceives innovation in three dimensions; the technical specifications of the product, its process and its service specifications. In this chapter, based on this framework, it is argued that focusing solely on technological distance, as it is done in most of the studies in this tradition, falls short of explaining a very important phenomena that many of the real world alliances reveal. In these cases there are strong synergies between the products of firms, independent of their technical knowledge endowments.

Agent based simulation is the methodology used in this chapter. The simulations aim to observe alliance and learning of firms which are randomly scattered at the start on a torus shaped space with positions defined by their technological and market address. Firms have different preferences when they are selecting partners, depending on the distance between them in both dimensions. After alliance, coordinates of firms in this space change, as well as they learn. In this way, inter-firm networks form and evolve. The relation between partnership distance preferences of firms and their final accumulated knowledge is investigated. Results are analyzed with reference to the networks that form during this process. Results show that firms which prefer close connections in one dimension, but distant connections in the other have higher returns. In other words, an alliance in which either market domain or technology domain is distant, but not both, proves to yield the highest performance. Accordingly, economies of scope can be seen briefly as the opportunity given to two firms having similar knowledge base and active in different markets to create alliance.

In the next section, the theoretical background is presented. Section 3.2 involves to the explanation of the model, including the analytic framework, assumptions and technical information on simulations. Section 3.3 presents results obtained. The last section is about discussions and interpretations of the model and the results.

3.1 Theoretical Background

The theoretical background of this chapter is based on complementarities and economies of scope in the light of Saviotti and Metcalfe (1984) model. In the next sub-section, the notion of distance within the context of complementarity between firms' resources are presented. Then, the Saviotti-Metcalfe model of innovation, which makes the basis of the simulation model used in this chapter, is presented. The last part deals with the economies of scope within the Saviotti-Metcalfe innovation model.

3.1.1 Complementarities and distances in firm resources

Complementarities of firms are measured by the concept of distance to each other in various literatures. One of the important literature which have used the term distance is found in the literature related to exploitation and exploration March (1991).

The literature which stems after after the seminal article of March (1991) on exploitation and exploration has different interpretations. The extensive literature review made by Li et al. (2008) on this subject results with a theoretical framework which aims to reduce the ambiguity around the concept of "exploitation and exploration". Li et al. (2008) argue that all authors, who have studied exploration and exploitation issues, agree that "exploration is the search for new knowledge, technology, competences, markets or relations, and that exploitation is the further development of existing ones". In the technological innovation literature, which refers to the work of March (1991), distant or local knowledge search is studied with respect to different dimensions. Li et al. (2008) group them in three dimensions; 1) cognitive, 2) temporal and 3) spatial. The cognitive dimension is measured according to the firms' familiarity to the newly searched knowledge. In practical terms, this is measured by using patent classifications[2] (Rosenkopf and Nerkar, 2001). The temporal dimension is related to the role of the time in knowledge search and the tension between exploration and exploitation. The creation of new knowledge by using recent new knowledge is temporal exploitation. Temporal exploration is the search of older knowledge to create new knowledge Nerkar (2003). The search of knowledge in spatial dimension take into account geographical distances between firms and the source of knowledge. According to Li et al. (2008) the degree of exploration/exploitation is defined by the distance of the knowledge search vis-à-vis to the firms' cognitive, temporal and spatial dimension of the knowledge space.

In other studies, it is claimed that knowledge has a substantial position in the complementarities between firms to make inter-firm alliances (Kogut

[2]Patent classification is also used in Chapter 5 to find out the knowledge diversity of firms and measure their knowledge boundaries.

and Zander, 1992; Spender and Grant, 1996). In this research tradition, complementarities of firms are measured either in terms of similarities or differences between firms involved in alliance. This measurement is generally carried out by the non-overlapping characteristics of products of two firms based on their differences (Gulati, 1995; Chung et al., 2000; Rothaermel and Boeker, 2008). Other similarities measures which are developed in the literature are based on the technological base similarities of firms (Mowery et al., 1998), overall innovative potential similarities (Rothaermel and Boeker, 2008), similar strategic groups which firms belong to (Nohria and Garcia-Pont, 1991), similarities in management practices (Lane and Lubatkin, 1998) and coherence in knowledge bases (Saviotti and Metcalfe, 1984).

In the literature, generally the market distance between firms is not extensively treated like the technological distance. Most often, a firm alliance is valued according to the additional willingness of consumers to pay for the resulting product. When technical competences of firms are not sufficient to release a product into the market, strategic alliances with firms which complement the missing resources are used. This combination of resources aims to provide a product combining the knowledge of firms which made the alliance. In previous studies through the contingency approach, which claims that the extent of resource complementarities between firms depend on product characteristics or the type of the project the alliance targets (Poppo and Zenger, 1998; Garrette et al., 2009). Based on this framework, firms are likely to take into account both their market complementarities and technical complementarities in forming alliances. While both market side and technology side of complementarities seem to be an important part of the partner selection process, there exist few studies, like Rothaermel and Boeker (2008), which take into account both dimensions together. The existing literature on strategic alliances largely underpins a joint consideration of the market side and technological side of complementarities between firms.

The model proposed in this chapter, which is using the technical and market dimensions, is inspired by the works of Saviotti and Metcalfe (1984). Authors make the distinction of technical, process and service characteristics of products.

3.1.2 The Saviotti-Metcalfe model of innovation applied to alliances

In Saviotti and Metcalfe (1984) framework, a product is represented in three dimensions which are technical, process and service characteristics. In this model, innovation is conceived either as a technical architecture or as a change in the range of services that products offer. In this chapter only two characteristics of the Saviotti-Metcalfe model are used to define an ICT product. Complementarity in firm resource has been generally measured by technological distance between firms. In this chapter, as the second dimension the market distance is added to the model. Both dimension are assumed to represent firm resource. The modified version of the Saviotti-Metcalfe innovation model is represented in two dimension. The boundary condition of this model imposes the use of a torus space which is found to be simpler to conduct agent based simulations compared to that of the surface of a sphere (Helbing and Balietti, 2011).

The first set of specifications are concerned with the internal architecture of products, whereas the second set of specifications are concerned with the use value to the consumers. These clusters of technical specifications and the clusters of service specifications are related to each other in complex ways (Windrum et al., 2009). Achieving desired service characteristics depends on solving problems related to the constraints imposed by technical specifications.

Figure 3.1 shows this framework extended to cover two potential partner firms. The technical specifications of products of firm i are given as $(T_{i1}, T_{i2}, \ldots, T_{in})$, and its service specifications are given as $(S_{i1}, S_{i2}, \ldots, S_{in})$. The mapping between technical and service specifications are given by (S_i, T_i) for each firm, which depends on the firm specific competences, and the nature of knowledge requirements of products. In this scheme, two dimensions of complementarities between the two firms are defined. The first one is given by c_t, which refers to the overlap of technological competences of two firms. The second dimension is given by c_s which refers to the overlap in the service specifications of the firms' products. In fact, contrary to what

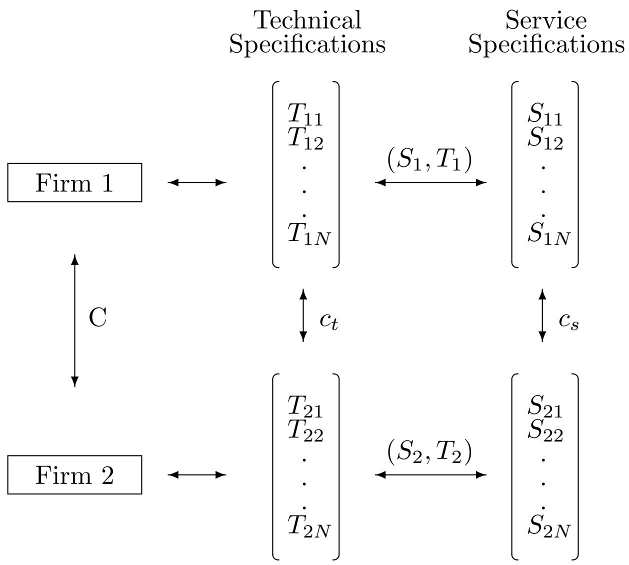


Figure 3.1: Saviotti-Metcalfe model adopted from Windrum et al. (2009) to the case of alliances.

one might think at first hand, these complementarities are independent from each other. Two firms can serve the same market domain, yet draw upon different technical competences. This is usually the case in complementary products, and services like real estate agencies and DIY retail chains which address the needs of home owners. At the same time, two firms can have very similar technical competences, yet apply these in different market domains. This case is usually valid in the ICT sector, where there are increased opportunities to reuse existing knowledge in different designs (Steinmueller, 2007; Lavie, 2007).

In this framework, the role of an alliance between two firms can be analyzed in two categories:

1. "Alliance between two firms can aim to redefine the mapping between service specifications and technical specifications of products, without a major alteration in the desired service specifications. In other words, while S_i does not change, (S_i, T_i) is altered. In this case, the features of the products which are "hidden" to the users are altered. This can

be because of an additional capability that firm i does not have, but which the other firm provides.

2. An alliance between two firms can aim to alter the service specifications S_i of a product, by adding desirable features to an existing product. In this case, (S_i, T_i) may or may not be altered, depending on the extent of modularity of the innovation"(Ozman and Özaygen, 2011).

The alliances which fall into the first category, redefine the existing mapping between service specifications and technical specifications. Usually, there will be some change in the service specifications, but not necessarily. This is the case when the alliance targets a modification in the production processes. The case which illustrates the first category of alliance is the alliance between Ford Motor Company and Pilkington Brothers Glass company. This alliance is made in the 1920s which resulted in the continuous processing of sheet glass. This procedure increased significantly the efficiency of producing large amounts of glass suitable for automobile manufacturing (Utterback, 1996). A similar example in software happened when Microsoft created alliances with GSM phone manufacturers to develop their operating system already used on PCs to use on mobile products. In these cases, a critical technological capability which a firm lacks, is complemented by other firms having the critical competences, thereby alliances change the existing design constraints. In this way, desired service specifications can be achieved through a redefinition of the mapping between technical and service specifications, so that (S_i, T_i) is replaced by $(S_i, (T_1, T_2))$. Note that, in this case, the alliance is motivated by the technical complementarities between firms.

In the second case, the additional service specifications of the targeted product shapes the extent to which firms are motivated to form the alliance, and their expected returns (S_+). An example is the alliance between Nike Inc. and Apple Inc. in 2006 for the production of smart shoes. An additional kit placed in the sole of the sports shoe permits the various performance measures to be recorded in the iPod of the user. In this case, while the service characteristics of the final product changed significantly, in the technical (or production) level, the only requirement was the addition of a modular pocket to the sole of the shoe, which did not require significant design changes in both companies' products.

In this case, an alliance between two firms adds new service specs to an existing product represented as $(S_i, n+1)$. The value of the added design feature, and consumers' willingness to pay for it, shapes the motivation behind the alliance. To the extent that the innovation is modular, meaning, the addition of $(S_i, n+1)$ does not affect the existing mapping between (S_i, T_i), limited technical overlap between two firms can be tolerated, yet the new product design creates significant value to the consumers. In this case, again there can be increased motivations to form an alliance between firms, regardless of their technological overlap, but because of a high market overlap.

3.1.3 Economies of scope

Panzar and Willig (1981) define economies of scope as a cost saving process from the combination of two or more production lines in one firm than to produce them separately. For the condition of two outputs y_1 and y_2, Teece (1980) gives the cost of joint production as;

$$c(y_1, y_2) < c(y_1, 0) + c(0, y_2) \tag{3.1}$$

According to Panzar and Willig (1981) economies of scope will result with multi-product firms. However, Teece (1980) argues that the conclusion given by Panzar and Willig (1981) is too strong; economies of scope is not necessary for cost saving by merging. Teece (1980) indicates that a multi-product firm can capture the economies of scope where the production of two or more products depends on the same know-how and when an indivisible specialized asset is a common input. In terms of Saviotti-Metcalfe model, then, this is the case where technical specifications are similar, but service specifications are different.

In ICTs, unlike the above definition of the economies of scope, the joint production is not restricted to a single firm but apply to all of the products incorporating interoperable components which are used in different combination targeting different markets (Steinmueller, 2007). The economies of scope is inherent to the software industry. In this work, the Saviotti-Metcalfe model is inspired to define firms in a knowledge and market dimensions and applied to alliances.

Based on this scheme on complementarities, the model that is presented below aims to explore which types of strategies have higher learning. The evolution of networks, and the accumulated knowledge among a heterogeneous population of firms are investigated. In this chapter, the alliance network and the accumulated knowledge are formed according to the perceived complementarities of firms in market and knowledge dimensions.

3.2 The model

The simulation starts with a population of firms located randomly on a torus surface. Each firm's position is defined by their technology and market addresses. These positions are denoted for firm i as m_i for its market position and k_i for its knowledge position. Torus surface is chosen so that firms can move in any direction.

Firms search for partners on this surface for partnerships. A body of empirical work on strategic alliances positions firms in some notion of space, and measure motivations behind alliances with respect to the distance between firms in the defined space. Some commonly used notions of space have been geographical space (Gomes-Casseres et al., 2006), cognitive space (Nooteboom et al., 2007; Schoenmakers and Duysters, 2006), social space (Gulati and Sytch, 2007) and strategic space (Gulati et al., 2000).

In this simulation the concept of distance is the projection of similarities of firms represented in one dimension. A close distance between two firms in one dimension shows that these two firms share similar knowledge base in that dimension.

In this model, it is conceived that learning is the return that the firm receives after an alliance. Learning depends on the distance between partners in two dimensions. The first dimension measures the technological distance and the second dimension refers to the market distance between two firms.

Firms are idiosyncratic in their choices; some of them may prefer close partners in the two dimensions, and some of them may prefer distant connections in the two, and others may fall in between. Firms are defined with two parameters (σ_i^m and σ_i^k) which show partner preferences of firm i depending on market and knowledge distance. For example, a firm which prefers close knowledge connections and distant market connections will have $\sigma_i^k \ll \sigma_i^m$[3]. These parameters are set randomly at the start of the simulation for each firm and do not change until the end of the simulation.

To distinguish firm types a scale (γ) between 0 and 1 is created:

$$\gamma_i = \frac{min(\sigma_i^m, \sigma_i^k)}{max(\sigma_i^m, \sigma_i^k)} \tag{3.2}$$

For firm i, equation 3.2 gives a result between 0 and 1 from which a scale on alliance preferences of firms are created. This scale is divided into three equal parts which gives three types of firms. These three types of firms are shown in tabular form in Table 3.1.

Table 3.1: Firm type and partner distance preferences ratio of firms γ.

Firm type	γ
I	[0.00; 0.33]
II	[0.34; 0.66]
III	[0.67, 1.00]

The first type represents firms which have a strong preference for close partners in one dimension and distant partners in the other dimension. By taking into account the concept of *cognitive distance* developed by Li et al. (2008) in relation to the exploration and exploitation litearture (March, 1991). It is defined in this chapter that exploration refers to experimentation with new alternatives and exploitation aims extension of existing competencies, technologies and paradigms March (1991). Firms which balance

[3] "Close" knowledge connections refer to the case where firm chooses a partner with similar knowledge endowments. Distant market connection refers to a case where products are different.

simultaneously the process of recombination of resources for exploitation and exploration are called ambidextrous firms (Tushman and O'Reilly III, 1996; Cantarello et al., 2012) and in this chapter are denoted as Type I firms. On the opposite end of the scale there are type III firms. Within this group there are pure exploiters but also pure explorers. Exploiters are firms which are creating alliances with firms which are close in both dimensions. On the other hand, explorer firms are searching for firms which are distant in both dimensions. These firms are searching for partners having competencies far from their own ones both in technological and market space.

Finally, the type II firms are between these two groups. They are neither exploiters/explorers nor ambidextrous. This group of firms do not have a distinct search for a partner which is similar or dissimilar to them.

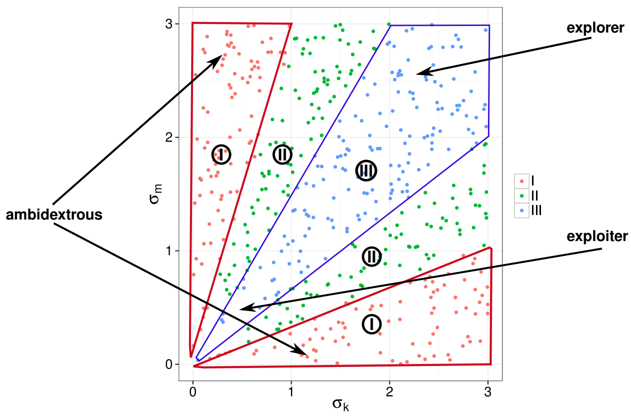


Figure 3.2: Firm types with respect to σ_m and σ_k.

The graphic representation of the three firm types is given in Figure 3.2. Grouping firms under three different types helps to isolate the effect of the alliance preference parameters on the accumulated knowledge of firms and their number of alliances.

Through a matching process, firms form alliances by forming pairs. The effect of performing an alliance is twofold; first they learn and second, they become closer to their partner (Baum et al., 2009). In each cycle firms try to form alliances, if any alliance happens then firms learn and move to their new location. Data are collected for each cycle; structure of networks that emerge, the relationship between firm preferences and system wide learning is analyzed.

To summarize, at the start of the simulation each firm i has two properties. The first one is its location on the torus surface, represented by (m_i, k_i). The second one is firms' preference for connections in market and knowledge spaces, given by (σ_i^m, σ_i^k). These two features are assigned randomly to the firms at the start of the simulation. After the start of the simulation, the next step is partner preferences; each firm evaluates all other firms as a possible partner. Then, if possible, according to the matching rules, firms create dyadic relationships. In the last phase, firms which have made an alliance move on the market and knowledge space to approach to their partner. This process repeats itself until the end of a simulation run. At the end of the simulations collected data are analyzed to understand the dynamics of learning.

3.2.1 Before alliance: Partner preferences

Two confirmed findings in the literature indicate that, firstly, the likelihood of an alliance between two firms is higher when their distance is at an intermediate level (Mowery et al., 1998). Secondly, an inverted-U relationship exists between firms' technological distance and their learning (Mowery et al., 1998; Gilsing et al., 2008; Schoenmakers and Duysters, 2006; Nooteboom et al., 2007). Moreover, this distance diminishes as firms made alliances more with each other (Mowery et al., 1998). The underlying logic in this construct is that, when firms are too close in the knowledge space, they have few to add to each others knowledge. However, when firms are too far they have difficulties to access each others knowledge base resulting with a limited learning.

According to this framework, an inverted-U relationship between learning and distance is assumed in this model. The optimal distance depends on firm's own perception on technology and market position of its partner. Deviations from the optimal distance will only have the effect of reducing learning and movement on the market and technology space of firms. If firm i finds its partner in the optimum distance according to its alliance preferences, it attains the maximum learning. These properties are satisfied with a second order function, given in Equation 3.3. This equation only represent learning of a firm i from its alliance with firm j and it is independent from the notions of market and technology for ease of representation.

$$\phi_{ij} = \beta(d_{ij} - \sigma_i) - (d_{ij} - \sigma_i)^2, \quad 0 < \beta, 0 \leq \sigma_i \leq 3 \tag{3.3}$$

ϕ_{ij} is the learning of firm i resulting from its alliance with firm j which is separated with a distance d_{ij}. σ_i defines distance preference of firm i to create alliances. Low σ_i denotes a preference of a close partnership while high σ_i represents a preference for a distant partnership, as shown in Figure 3.3. β defines the knowledge type prevalent of the system; low β relates to the tacitness and high β to the codification of the knowledge. In other words, when β is low, knowledge is not transferred easily and learning is less.

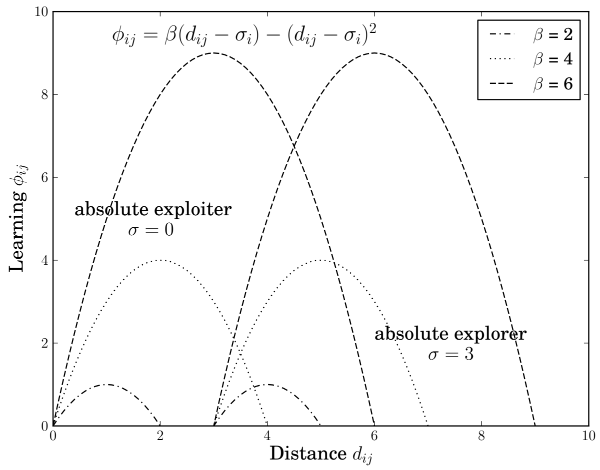


Figure 3.3: Learning (ϕ_{ij}) of firm i with respect to its distance (d_{ij}) to firm j in dominant tacit knowledge ($\beta = 2$) or codified knowledge ($\beta = 6$) environment. Firm characteristic is given by σ_i. In the left $\sigma_i = 0$ is absolute exploiter, and in the right $\sigma_i = 3$ represent absolute explorer firms.

In Figure 3.3, learning (ϕ_{ij}) of firm i is given as a function of its distance to firm j. For each β, two different curves representing two extreme values of σ are given. At the start of the simulation, randomly set σ for each firm permit to model the heterogeneity of firms in terms of their preference to create alliance according to their distance with their possible partners. Low σ corresponds to exploiter and high σ to explorer firms. Exploiter firms' curves are those on the left side and explorer firms' learning is given in curves which are at the right.

Each firm is characterized by a different σ_i pair, one representing preferences for the technological dimension (σ_i^k) and the other for the market one (σ_i^m). For each σ_i there is an optimal distance for firm i which maximizes its learning in one of the two dimensions. The distance between firms in the market and knowledge dimensions are given by d_{ij}^m and d_{ij}^k, and they are simple Cartesian distances taken separately in both dimensions:

$$\begin{aligned} d_{ij}^m &= m_i - m_j \\ d_{ij}^k &= k_i - k_j \end{aligned} \tag{3.4}$$

As firms are located on a torus shaped space, distance between two firms should be measured with the minimum length between them.

The learning (ϕ_{ij}^e) that firm i expects from its alliance with firm j, is the sum of expected learning in function of their distance in market and in knowledge space. This relationship is given bellow;

$$\phi_{ij}^e = \phi_{ij}^{em} + \phi_{ij}^{ek} \tag{3.5}$$

where the expected learning from the market and technical space is given as follows;

$$\begin{aligned} \phi_{ij}^{em} &= \beta(d_{ij}^m - \sigma_i^m) - (d_{ij}^m - \sigma_i^m)^2 \\ \phi_{ij}^{ek} &= \beta(d_{ij}^k - \sigma_i^k) - (d_{ij}^k - \sigma_i^k)^2 \end{aligned} \tag{3.6}$$

then the expected learning of the firm i from its alliance with firm j is;

$$\phi_{ij}^e = \beta(d_{ij}^m - \sigma_i^m) - (d_{ij}^m - \sigma_i^m)^2 + \beta(d_{ij}^k - \sigma_i^k) - (d_{ij}^k - \sigma_i^k)^2 \quad (3.7)$$

In short, according to the equation 3.7, expected learning for firm i from its alliance with firm j has two components shown on the RHS. First, expected learning due to market complementarities, $\phi_{ij}^m(d_{ij}^m)$; second, learning expected from knowledge complementarities $\phi_{ij}^k(d_{ij}^k)$. In equation 3.7, σ_i^m and σ_i^k are alliance distance preference parameter of firm i in market and knowledge domains respectively.

3.2.2 Matching

The matching process that is used in this chapter is inspired by the Gale and Shapley (1962) matching process, which has been previously used in agent-based simulations by Ozman (2005) and Cowan et al. (2007). Originally, Gale and Shapley (1962) matching is for the symmetric case in which both for i and j there is not any alternative pairing. In the model used in this chapter, the mutual learning expectations are not symmetric i.e. what firm i expects from its alliance with j, and what j expects from i could be different due to the different distance preferences among firms.

Based on the equation 3.7, each firm calculates its expected learning from alliance with the other firm. In this process, two firms form a partnership, if and only if the mutual learning expectations are higher than the rest of the available partners.

3.2.3 After alliance

It is assumed that firms come closer to each other in the industry space after a partnership (Baum et al., 2009). The movement of the firm is the reflection of its learning process. For the firm i the expected learning ϕ_{ij}^e from its alliance with j becomes the realized learning ϕ_{ij}^r. The learning process is affected by the IPR regime. In a tight IPR regime learning is repressed through tight licensing and patenting practice. In this model the IPR regime is represented with α which affects the movement of firms after alliance. The new location of firm i, after its alliance with firm j is given in Equation 3.9. The highest, optimum learning that the firm i can obtain is as follows;

$$\begin{aligned} S_i^m &= \left\{ \phi_{ij}^m(d_{ij}^m) \quad | \quad \frac{\delta(\phi_{ij}^m)}{\delta(d_{ij}^m)} = 0 \right\} \\ S_i^k &= \left\{ \phi_{ij}^k(d_{ij}^k) \quad | \quad \frac{\delta(\phi_{ij}^k)}{\delta(d_{ij}^k)} = 0 \right\} \end{aligned} \tag{3.8}$$

such that the new position of the firm i after alliance with firm j is;

$$\begin{aligned} m_{it} &= m_{it-1} + \alpha \frac{\phi_{ij}^m(d_{ij}^m)}{S_i^m}(m_j - m_i) \\ k_{it} &= k_{it-1} + \alpha \frac{\phi_{ij}^k(d_{ij}^k)}{S_i^k}(k_j - k_i) \end{aligned} \tag{3.9}$$

Firms move toward their partners after the alliance with a system wide factor given as α. This factor is also multiplied by the ratio of the success of the firm which are $\phi_{ij}^m(d_{ij}^m)/S_i^m$ or $\phi_{ij}^k(d_{ij}^k)/S_i^k$. If the firm finds a partner at a distance d such that $\frac{\delta\phi_{ij}^k}{\delta d} = 0$ and $\frac{\delta\phi_{ij}^m}{\delta d} = 0$ where the firm i attain the maximum learning (S_i^m or S_i^k), the movement will be proportional to α creating the maximum movement for the firm i. If the partner is not located at optimum distance d, then the movement will be reduced proportionally to maximum learning. Therefore, high α represents tight IPR regime and low α is used to represent weak IPR regime. The movement is the longest when the partner is at the optimum distance. If the partner is too far or too close, the movement will be small.

At the end of cycle n, accumulated knowledge (Φ_i) of firm i is the sum of its learning obtained from previous alliances, given by the Equation 3.10. Table 3.2 gives the explanation of all variables used in this chapter.

$$\Phi_i = \sum_{0}^{n} \phi_{ij}^r \tag{3.10}$$

Table 3.2: Definition of variables used in the simulations.

Φ	accumulated knowledge for a firm from cycle 1 to n
ϕ	learning in one cycle from a partnership
β	knowledge type (from tacit to codified)
α	IPR regime (from weak to strong)
γ	firm types [0,1], $\gamma(\sigma_m, \sigma_k)$
σ	firm characteristics [0,3], (from exploiter to explorer)
d	distance between two firms
m_i	market position of the firm i
k_i	knowledge position of the firm i

3.2.4 Assumptions

It is assumed that firms select partners based on their perception of distance. This is a realistic assumption as most studies reveal. But a unique optimal distance which is valid for all firms is not given. In other words, it is assumed that how each firm defines "closeness" is different. A heterogeneous population of firms is assumed. Firms have different criteria in selecting partners because firms' distance preference parameters are set randomly.

One of the important assumptions of the model is that, alliance distance of preference parameters of firms are kept constant throughout a simulation run. At first this might seem as a strong assumption. Nevertheless, previous studies found that, the stage in the life cycle of an industry determines the extent of connections between firms (Pyka, 2000; Nesta and Mangematin, 2002). Particularly in the beginning of a life cycle, relations are denser, and exploratory alliances are dominant. As the dominant design emerges, firms converge to a particular design (they come closer in the technology and market space), and relationships are predominantly exploitative, to deepen competencies (Suárez and Utterback, 1995). Therefore we prefer to fix the distance preference parameter throughout the simulations, and characterize each firm by its connection strategy. In this case, the model should be interpreted in a particular industrial context, and analysis is from the beginning of an industrial life cycle to a more mature phase.

In this model, firms also make breakthrough innovations (Hargadon, 2003). This process follows a Poisson distribution and firms realizing a breakthrough change randomly their position within the market and knowledge space.

3.2.5 Simulations and parameters

Simulations are conducted for various α (IPR regime) and β (knowledge type). These values are system wide and do not change during one simulation. Here, α determines the IPR regime. There are two opposite IPR regimes, knowledge diffusion is easier in weak IPR regime and difficult in strong IPR regime. In weak IPR regimes, imitation between firms is simple without much legal issues encountered. An example of a weak IPR regime is the weak patent regime in which patents have a higher probability of being overturned and easily circumvented (Anton et al., 2006). Low α is used to simulate weak IPR setting and high α is used for the strong IPR setting. High α causes firms faster displacement within the knowledge and market spaces. β determines the maximum learning; as β increases the dominant knowledge type of the system moves from tacit to codified knowledge. On the other hand, if the dominant knowledge type is codified then β increases. For this chapter various simulations are carried out with $\alpha \in \{0.05, 0.1, 0.2\}$ and $\beta \in \{2, 4, 6\}$. Thus in total, for each α and β pair, nine simulations with different settings are conducted.

Five different simulations for each α and β combinations are done and given results are average values which are obtained from these five simulations. Each simulation consists of 500 cycles. Simulations start with a population of $N = 100$ firms. The breakthrough process follows a Poisson distribution with parameters $\lambda = 0.1, k = 500$. The coordinates of firm i in its entry is drawn from a uniform distribution such that $(m_i, k_i) \in [0, 20]$. The maximum real distance between firms can not be over than 10 units on both axes because firms are positioned on a torus like surface. The initial distance preference parameters are given as $(\sigma_i^m, \sigma_i^k) \in [0, 3]$ which are uniformly distributed.

During the simulations only the number of firms, occurrence chance of breakthrough values and cycle numbers are kept constant. Firms' position on the technology, market space and firms' alliance preferences (σ_m and σ_k) are set randomly at the start of each simulation run. In the model, there are only bilateral links in a single period, but after 500 periods, a network through the accumulation of relations is obtained. Figure 3.4 shows the algorithm of the model in the form of a flowchart.

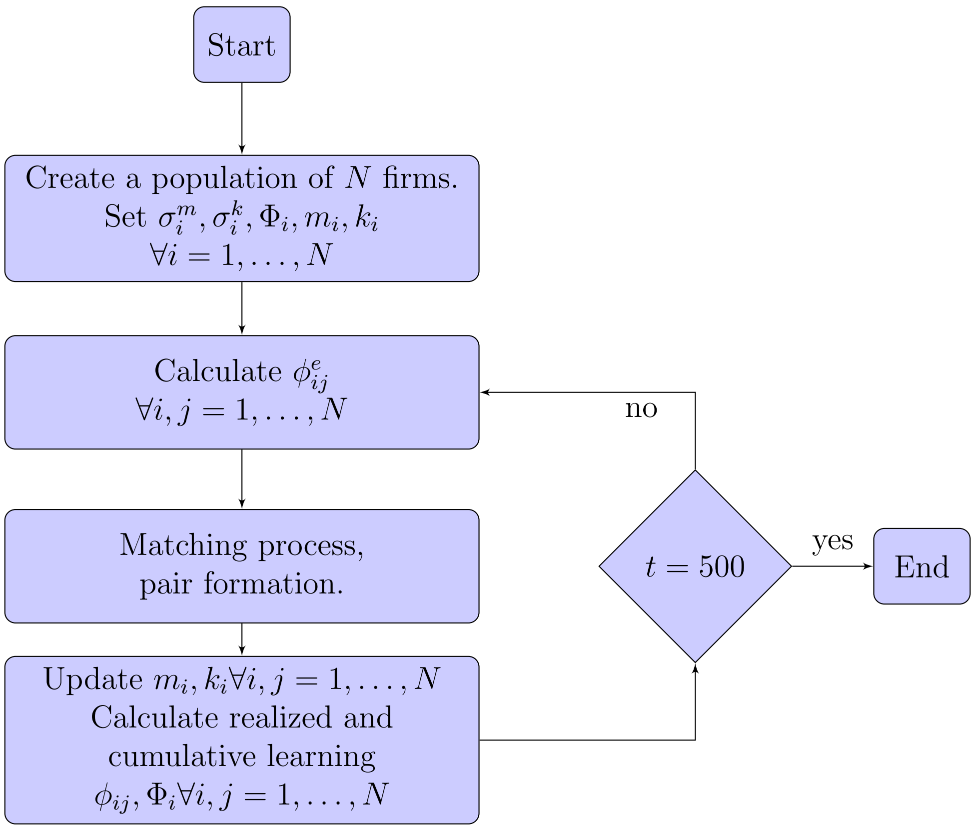


Figure 3.4: The algorithm of the model.

3.3 Results

Throughout the whole simulation all firms increase their accumulated knowledge. Figure 3.5 shows the effect of different knowledge types (β) on firms' average accumulated knowledge within different IPR regimes (α). As expected, and shown in the Figure 3.5, as the knowledge codifies, i.e. as β increases, the accumulated knowledge also increases. It is also observed that, as the IPR regime becomes stronger, i.e. as α increases, the accumulated knowledge decreases.

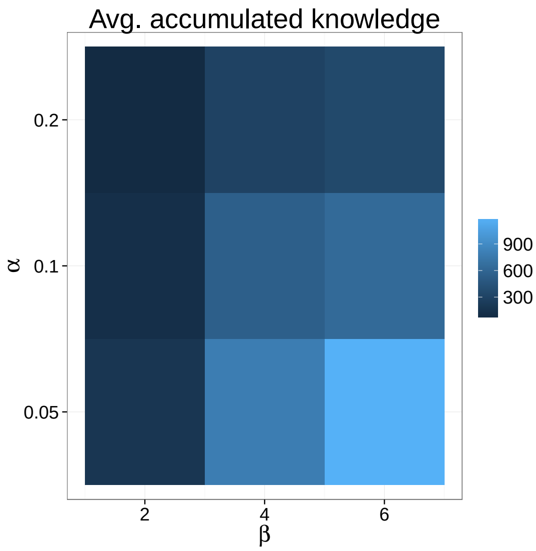
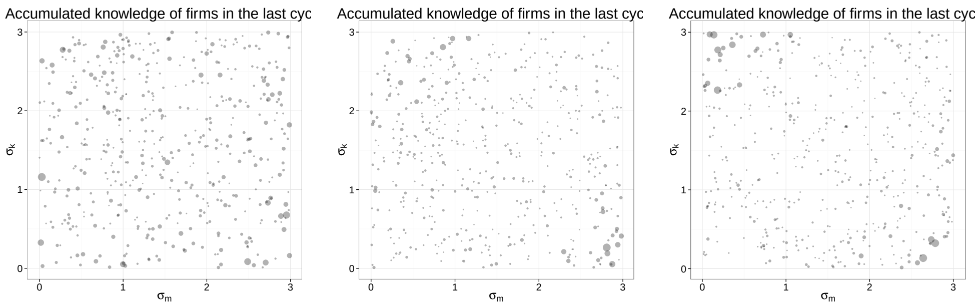


Figure 3.5: Average accumulated knowledge after 500 cycles.

There are three types of firms. Type I firms are ambidextrous, those which are inclined to create alliances with a partner being close in one of the market or knowledge dimensions and distant on the other dimension. That is, type I firms prefer to create alliances with firms which are explorer in one dimension and exploiter on the other one. Type III firms are the ones which prefer to create alliances with firms which are either exploiter or explorer in both dimensions. Type II firms are firms which are between these two types of firm.

The distribution of accumulated knowledge of firms according to the σ_m and σ_k values at the end of 500 cycles are given in Figure 3.6. One simulation with distinct α and β parameters are made of five different runs. In each run σ_i^m and σ_i^k values of firms are set randomly. In Figure 3.6 each point is a firm, and in each figure there are 500 firms obtained from five different



(a) $\alpha = 0.05, \beta = 6$ (b) $\alpha = 0.1, \beta = 6$ (c) $\alpha = 0.2, \beta = 6$

Figure 3.6: The distribution of accumulated knowledge of firms according to the σ_m and σ_k values of firms at the end of 500 cycles, all five simulations final values are superimposed.

simulation runs. Point sizes are proportional to the accumulated knowledge of firms. Figure 3.6 reflects the result of a system which is dominated by codified knowledge ($\beta = 6$) but with varying IPR regime parameters (α). Figure 3.6(a) shows accumulated knowledge of firms under codified knowledge and weak IPR regime. Within this setting average accumulated knowledge difference within different firm types is not as much as the one shown in Figure 3.6(c). In codified knowledge regime under strong IPR parameter α, type I firms have higher accumulated knowledge, i.e. ambidextrous firms learn more in a rich economies scope environment like the software industry. Accumulated knowledge differences based on firm types are given in Figure 3.7.

At the end of 500 cycles, average learning of type II and type III firms follows more or less the same trend, however, type I firms have an obvious lead in average learning. These results show that ambidextrous firms have always higher learning in any IPR regime and knowledge type. Moreover, in rich economies of scope such as software industry, type I firms, which find partners from different environments, profit more from this setting. To show this result, the average of type II and type III firms' average accumulated knowledge is subtracted from type I firms' average accumulated knowledge. Results, obtained for different IPR regime and knowledge settings, are given in Figure 3.7. It is found that the difference between type I firms accumulated

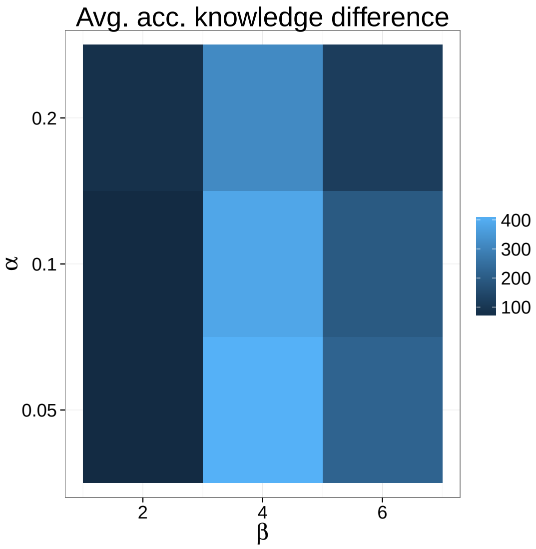


Figure 3.7: Average accumulated knowledge difference of type I firms compared to the average of type II and III firms after 500 cycles.

knowledge and those of the other two do not change linearly as the system's dominant knowledge type (β) changes. It is found that there is a transition phase related to the dominant knowledge. In this phase, the advantage of type I firms is increasing then this difference decreases when the knowledge is more codified. This transition is shown in Figure 3.7 at $\beta = 4$ for all different IPR regimes (α). As the IPR regime weakens this difference is increasing. In other words, ambidextrous connection strategies are relatively highest payoff in intermediate β.

According to the initial model specifications, the only way to increase the accumulated knowledge is by creating alliances. This is why, the number of alliances of firms is positively correlated with their accumulated knowledge. However, learning is also related with the distance to the partner firm and especially whether it is within an optimum range. In other words, the more a firm can find partners within an optimum distance range, the more it learns, and thus its accumulated knowledge increases. At the start of each simulation all firms are evenly distributed on the torus surface. As the number of cycle increases, depending on the IPR regime (α), positions of firms on

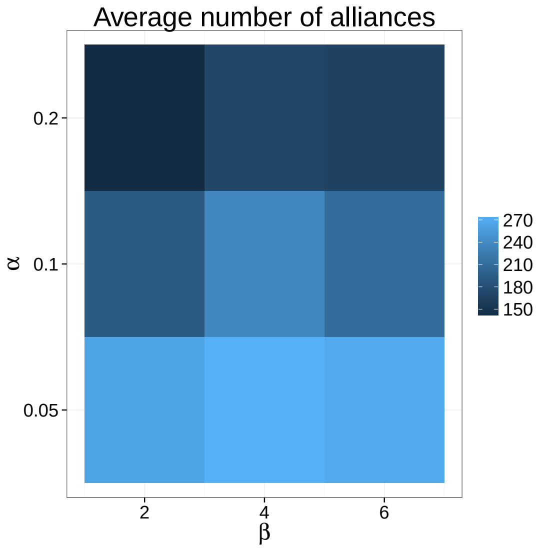


Figure 3.8: Average number of alliances after 500 cycles.

the knowledge and market spaces change and the possibility to find partners depends on typology of firms based on σ_m and σ_k parameters. The average number of alliances created under different IPR regimes (α) and knowledge (β) settings are given in Figure 3.8. In this figure, there is a transition in which the number of alliances increases at $\beta = 4$ but under tacit and codified knowledge settings the number of alliances is lower.

The average number of alliances per each firm type, in each cycle, shows that type I firms are making continuously the highest average number of alliances throughout the whole simulations. The difference of average number of alliance between type I firms and others is given for all simulation parameters in Figure 3.9. The difference between firm types on the number of alliance changes linearly with respect to the codification of the knowledge and also to the change in the IPR regime. As the knowledge codifies and the IPR regime weakens the difference between type I firms and the average of the type II and type III firms increases. Type I firms are the ones which increases its number of alliances within the economies of scope as the IPR regime weakens and the knowledge codifies.

The total number of created alliances by the end of 500 cycles are given

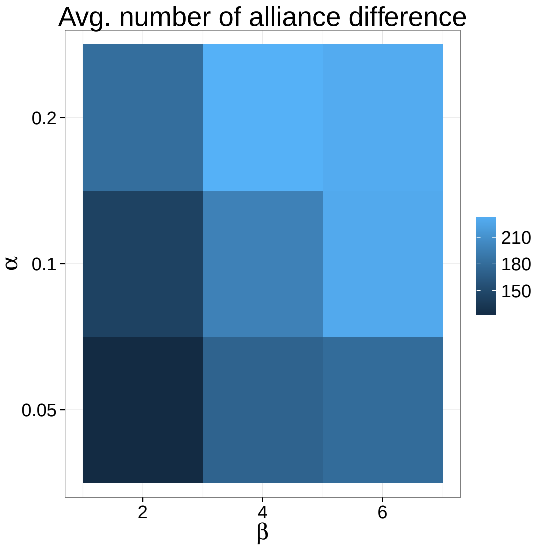
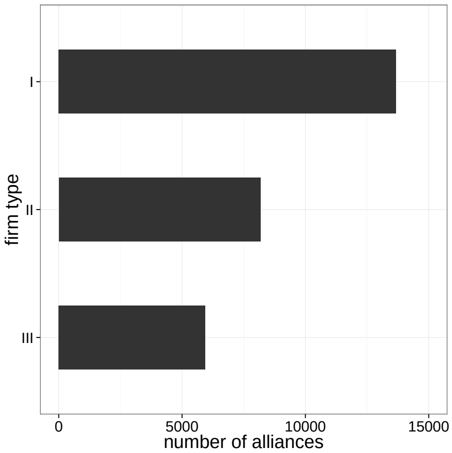


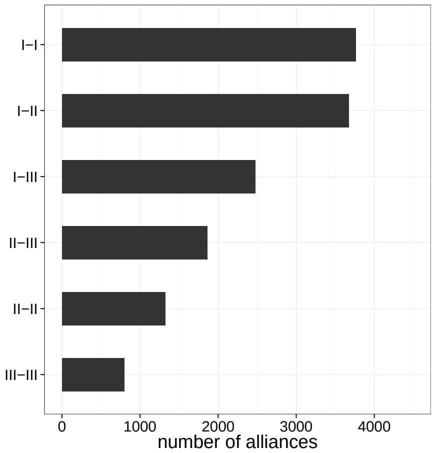
Figure 3.9: Average number of alliances differences between type I firms and others after 500 cycles.

in Figure 3.10(a). Type I firms are creating the highest number of alliances then comes type II and the last one is the type III. This property is even more pronounced under strict IPR regime, i.e. for higher α, as it is discussed above. The total number of alliances created between different firm types are given in Figure 3.10(b). This graphic shows that type I firms prefer similar type of firms whereas other firms have less number of alliance within their group. These results show that ambidextrous firms are in advantage in the number of alliances as well.

During the simulations in every cycle, graphs on the position of firms on the torus surface is obtained. These graphs, which are not provided, show that there are two different phases during one run of a simulation. First, firms are scattered evenly on the market and knowledge space. Then with alliances, firms generally create few clusters on this space. Then, during the second phase alliances continue but it is either inter-cluster or intra-clusters. From these images it is clear that clustering occurs between cycle 100 and cycle 200, depending on other simulation parameters. During the second phase the rate of the accumulated knowledge decreases, in average firms learn less



(a) Number of alliances by firm types.



(b) Number of intergroup alliances.

Figure 3.10: Number of alliances created between different firm typologies ($\alpha = 0.05$ and $\beta = 4$).

compared to the rates obtained during the first phase of the simulation. After clustering, differences according to the firm typologies became obvious.

To summarize; the accumulated knowledge according to different IPR regimes (α) and knowledge settings (β) obtained after 500 cycles is given in Figure 3.5. Figure 3.7 shows the average accumulated knowledge difference of type I firms compared to the average of type II and III firms after 500 cycles. This situation shows that the change in dominant knowledge type affects ambidextrous firms most, while other firms which are either explorers or exploiters are slow to catch this transition. As the system moves toward a more codified regime then the accumulated knowledge for type II and III firms approaches to that of the type I firms. The number of alliances also shows that type I firms have the lead on type II and type III firms. As the exchange of information becomes easier with the codification of the knowledge, this difference decreases but the type I firms are always having the highest accumulated knowledge.

3.4 Discussion of the results

Similar designs are used in different application needs within the knowledge regimes with rich economies of scope. This production is found in ICTs with product diversification as one of the main result (Steinmueller, 2007). The product diversification and the entry into new markets is obtained through co-operation and alliances. Some of the firms serve completely different markets, yet they have a similar knowledge base. This case is common in ICT industries, where reusability of knowledge in different designs is a common trait. Another example would be firms which are in different geographical locations and through alliances they explore different markets. In this chapter, a model is designed to simulate these aspects through alliances in different IPR regimes and knowledge settings. The aim is to understand the accumulated knowledge of firms in relation to their alliance preferences in knowledge and market dimensions.

Results show that, the number of alliances and accumulated knowledge increases as the knowledge codifies, and also as the IPR regime weakens. The organizational learning literature traditionally focuses on the trade-off between investment of firms in exploration and exploitation. While exploration refers to experimentation with new alternatives, exploitation aims at refinement and extension of existing competencies, technologies and paradigms (March, 1991). It is now widely accepted that these two dimensions of organizational learning are not substitutes but complements each other. This is called the ambidexterity hypothesis (Tushman and O'Reilly III, 1996; Lavie and Rosenkopf, 2006). In other words, for higher innovative performance firms should have the capability for, and invest both in exploration and exploitation.

During the initial stage of the simulations, firms are largely scattered in the technology and market space, and where exploratory alliances are prevalent in the industrial system. As a dominant design emerges, it is accepted by firms as the standard, which marks the beginning of a period in which the environment becomes more stable. Relative weight of exploitative alliances increases in the industry as firms converge to each other in terms of their technologies and products (Rowley et al., 2000).

Results obtained from the simulations show that a successful firm is explorer in one dimension, and exploiter in the other dimension under different IPR regimes and dominant knowledge types. During the transition from tacit to codified knowledge, ambidextrous firms have much more advantages compared to other firms. These results are obtained without incorporating context specific parameters about knowledge and markets, but just with respect to the relative positioning of firms and their alliance preferences in the knowledge and market space.

Lavie (2007) shows that software firms which create alliances with partners having abundant marketing and financial resources obtain positive market performance returns. However, she also shows that networking based on technology and human resource are not contributing to the market performance of firms. According to the author this positive return is due to the nature of software firms which are specialized in their proprietary technology and their human assets. It could also be argued that these firms are also realizing profit according to the economies of scope which is prevalent in software industry. Results from simulations are similar to the results obtained by Lavie (2007). However, this study do not give any information on the alliances of firms according to their technological distance. Results obtained from the simulations show that learning is higher for ambidextrous firms. Moreover, the results of the simulation made in this chapter replicate an empirical study of Wiersema and Bowen (2008) on corporate strategy moves of firms on their degree and scope of international diversification. Authors find that firms are facing a trade-off when they expand via product or geographic, i.e. market, diversification. Firms should carry out one of these things at a time.

One of the difficulties in simulation studies is concerned with the ability to interpret abstract notions from real life examples. For this model it will be useful to present some real world examples in which these results can be interpreted. In some cases, a strategic alliance between two firms from completely different market and technology domains can result in a completely new design. This is the case when there is strong complementarities between the markets of two firms, and the aim of the alliance is to exploit these complementarities. Some of the examples of alliance between firms which are close in market, distant in knowledge are; the alliance between Nike Inc. and Apple Inc. in 2006 for the production of smart shoes; the alliance between the publishing company Condé Nast and software company Adobe Inc. for digital magazines can be cited. Both of these alliances aim at a completely new product/service by combining their competences. These firms correspond to explorers in their technology distance and also in their markets.

Results of the simulations show that alliances between firms which are very close in one dimension and distant on the other dimension are very frequent. These alliances usually occur to strengthen/sponsor a certain standard in the industry, especially when there are strong network effects. Majority of the alliances in the real world correspond to the cases in which firms complement each other in different domains. A very common example is the alliances of software firms with different hardware manufacturers. Generally, software should be tailored specifically for different hardware platforms. Software firms keep the base source code but make various modifications to the software in order to run on different platforms and for various markets. For this purpose software firms make alliances with firms having an experience on specific platforms and specific markets. Two examples are; the development of operating systems and video game software. In the first example, an operating system (OS) runs on a hardware platform. Either proprietary or FLOSS, various OS run on hardware platform such as PCs, tablets, phones and so on. These three hardware platforms have generally different producers which have different technical and market competencies. Software is modified with hardware producers. The complementarities between market and technological domains provide increased opportunities for firms to make alliances. It is shown that ambidextrous firms have higher learning when they leverage rich economies of scope under various knowledge and IPR regimes.

As a results, in all IPR regimes and knowledge settings, it is found that ambidextrous firms i.e. firms which prefer similar partners in one dimension, and different partners in the other dimension are better off compared to other firm types. It is also found that in a transition from tacit to codified knowledge, ambidextrous firms increase their accumulated learning compared to other types of firms. As knowledge gets more codified; the relative advantage of ambidextrous firms are reduced. Other types of firms also increase their accumulated knowledge which results with a decrease of this difference.

Chapter 4

Analyzing modular technologies

In the previous chapter, effects of alliance preferences on learning and networking of firms have been investigated. Results showed that firms which have an alliance preference for partners in close market or technology dimension and distant on the other dimension had higher accumulated knowledge. This result is applicable to software industries as the software per se allows its application in different areas. Moreover, software technologies are also very open to create modular products. Modularity is a concept which is applicable to assembled products. In software technologies, modularity improves the reuse of the code in different products which contributes to the rich economies of scope found in the software industry. Modularity is also evaluated in this thesis as one of the structural characteristics of software technologies.

Research on modularity dates back to Simon (1962) who studied modularity within the research field of complexity. Theoretical work on modularity in software engineering has been carried out by Parnas (1972) who also contributed to the theory of object-oriented software development which aims to improve software development process. Modularity helped many companies to overcome the increasing technological complexity (Baldwin and Clark, 1997). Also, modularity within innovation studies is generally examined under product architecture (Ulrich, 1995) and its effect on organizational structure (Sanchez and Mahoney, 1996). One of the early statement on modularity, based on a research made on computer hardware and stereo systems, argues that modularity increases specialization at the firm level (Langlois and Robertson, 1992). There are two competing views on the innovativeness of firms according to their knowledge boundaries. The first one argues that diversified firms are more innovative through cross-fertilization, and the opposing view is that specialized firms which make R&D in few areas are more innovative (Brusoni et al., 2001). Consequently, the question that arises is whether modularity is moving firms into an increasing specialization or diversification?

This chapter and the next chapter aim to find whether firms which contribute to the development of a distinct software technology are specialized or diversified. Moreover, it is aimed to find the effect of increasing modularity on organizational structure of the software industry. The research questions of these chapters are:

- Do technological achievements in a highly modular software technology originate from firms which are diverse or which are specialized?
- What are the average knowledge boundaries of firms contributing to the development of a software technology?

This and the next chapters are complementing each other. In this chapter, theoretical background and methodological tools, which will help to answer the research questions, will be presented and discussed. In the next chapter, the above questions will be treated for the video indexing technology, a software technology which helps to translate automatically content of video files into semantic meaning. This technology helps to ease the search process on video files similar to the method conducted for text files.

In order to answer the research questions, it is necessary to understand how to depict and analyze technological achievements in software technologies. In order to accomplish this, patent analysis will be used since patents represent the best archive on inventions and cover virtually every area of innovation in most developed countries (Jaffe and Trajtenberg, 2005). This research is delimited with information obtained through patent analysis but expert help is also obtained. The aim of these two chapters, by using patent analysis is, first to determine those firms which contributed to the development of the video indexing and its modules with important technological achievements. Then, the technological diversity of these firms will be compared to that of the industry as a whole. This comparison will help to understand whether technologically diverse or specialized firms have made technological achievements and how the knowledge boundary of firms in software technology such as video indexing and its modules is prone to change.

Section 4.1 below is a review on modularity theory which also includes discussions on the effects of modularity on knowledge boundaries of the firm. In Section 4.2, methods deployed to locate important patents within a specific technology will be presented. The last section will focus on patent citation network analysis which will be used in the next chapter to answer the above research questions within the context of video indexing.

4.1 Modularity

To illustrate modularity Simon (1962) gives the example of watches as complex systems and portrays two watchmakers; Tempus and Hora. One of these watchmakers becomes successful by assembling modularly designed watches, the other in contrast becomes bankrupt due to his watches not being constructed with sub-assemblies. Modularity at its most abstract level is the degree to which a system's components can be separated and recombined (Garud and Kumaraswamy, 1995). A "complex system" is hierarchical and nearly decomposable. It is the combination of different parts that have non-trivial interactions. A complex system could be analyzed or created by decomposing it into its subsystems, and recursively its subsystems could also be analyzed in the same way. Complexity is formed through pure random interaction and evolution of simple systems. *"How complex or simple a structure is depends critically upon the way in which we describe it"* (Simon, 1962, p.481).

Difficulties linked to the increasing complexity of the knowledge that underlies products and the speed of technological change are worked out by modular design and production (Baldwin and Clark, 1997; Schilling, 2000).

The visible and the hidden information of a module are two core concepts which make modular design and production efficient (Baldwin and Clark, 1997). The hidden information of a module is its internal functioning mechanism which does not go beyond the local module. This gives the opportunity to an off-site facility in which the design and the production of modules, as far as it obeys the visible parts, are constructed. The visible portion is the architecture, interfaces and standards that a module contains in order to work consistently within an assembled product. These two concepts are not restricted to manufactured goods. The case of software and modularity has been developed by Parnas (1972) and presented in Section 4.1.3.

Modularity is used as a tool to overcome the complexity of knowledge and needs new management practices. Recommendations of Baldwin and Clark (1997) to managers is to have more insight on products designed and produced both in and out of their firms, so that they could take pertinent decision regarding the innovation management of the company and its position in the market. On the other hand, modularity is not a simple remedy, inventors could face "*pathological pressures for modularity*" (Fleming and Sorenson, 2001, p.1036).

Architecture and interfaces are crucial in modular products. Once interfaces and architecture are set, there is not much possibility for a radical innovation. Components combine easily with less system wide efficiency resulting with incremental innovations (Fleming and Sorenson, 2001). Firms can achieve high performance by reusing some components and substituting others. But according to Garud and Kumaraswamy (1995), economies of substitution create a technological change process which is neither incremental nor radical but include both of these characteristics.

Langlois and Robertson (1992) argue that modularity has benefits on the demand-side as well as on the supply-side. On the demand-side, consumers can fine tune, customize a modular product to fulfill their special needs. On the supply-side, modularity helps independent product developers to learn rapidly by trial-and-error during the early stage of the technological development. Langlois and Robertson (1992) continue to suggest that the result of the modularity is the formation of vertical and horizontal disintegration due to the rent appropriation of innovators. According to the authors, this disintegration process is contradictory to the argument that rent appropriation could be achieved by vertical integration, i.e. by acquisition of firm (Teece, 1986). This would facilitate incentive alignments and control over the firm producing specialized complementary assets.

Research on modularity has been conducted in various areas. In this chapter, modularity is treated under two main topics. The first one is the product architecture in which modularity is discussed mainly within product development and manufacturing. The second part deals with the effect of modularity on organizations. In relation to the theme of this thesis a third section is added which summarizes modularity within the context of software.

4.1.1 Product architecture

Ulrich (1995) defines modularity as a one-to-one mapping between elements of a product and physical components. Modularity helps firms to design various products with incremental innovations and few changes in the design. Fleming and Sorenson (2001) argue that if a firm has difficulties to release products from R&D to the market due to some technological complexities, the firm could use modular off-the-shelf components and release products that have more mature technology. Moreover, modularly upgradable systems can help firms to reduce product development time, reduce cost and provide customers with continuity (Garud and Kumaraswamy, 1995). The positive aspect of the modular architecture on product development has been illustrated as in the case of Sony Walkman (Sanderson and Uzumeri, 1995), Black&Decker power drills (Utterback, 1996), IBM System/360 computer series (Baldwin and Clark, 1997) and hardware and software used by Sun Microsystems (Garud and Kumaraswamy, 1993).

Modularity does not bring any efficiency if the level of interdependency between modules is high. The degree of interdependence among system components affects the success of the whole system. Modules could be easily replaced or changed by others if there is a low level of interdependence among them, but as the interdependence rises, it becomes very difficult to find a useful combination among modules (Fleming and Sorenson, 2001).

According to Baldwin and Clark (1997), there are two different parties within modular production. The first one is the architect who is the system integrator. The architect has to attract module producers by convincing them that the designed architecture is viable. The second one is the module producer who has to master the module production and enter into the market very rapidly. Upon the entry, the module producer has to move into another market or increase the performance of the module.

The distinction of system integrators and module producers is blurred with the maturity of the technology and the standardization. Product standards of modular production have a significant impact on the organizational structure of the industry. In the early phases of technological development there is a high degree of uncertainty. After the emergence of a dominant design (Suárez and Utterback, 1995) which is set by a few firms, standardization of parts of modular systems create the interchangeability across firms (Langlois and Robertson, 1992). As in the case of the bicycle modules, the emergence of worldwide accepted standards leads to a fragmentation of the industry with each firm operating independently from each other (Galvin and Morkel, 2001). Moreover, this fragmentation causes the lack of communication among firms with the difficulties to change the product architecture. Loose ties among module producers enforce modularity of the product thereby shaping the organizational structure of the industry.

4.1.2 Organizational structure

As modularity affects firm organization, it also shapes the relationship between firms in accordance with the development phase of the industry and the technology. Modularity has an important impact on innovative activities which influences the organization of design, knowledge management and learning activities within a firm (Garud and Kumaraswamy, 1995; Sanchez and Mahoney, 1996; Baldwin and Clark, 1997). Modular product architecture enables decentralized production resulting with loosely coupled, flexible and modular organizations with a simplified internal coordination (Sanchez and Mahoney, 1996; Schilling, 2000).

Baldwin and Clark (1997) argue that modularity increases the pace of change and transforms relations between firms. Langlois and Robertson (1992) describe the early stage of development during which the technology is rapidly changing with a high degree of both technical and market uncertainty. During this early development stage a decentralized production system allows the introduction of new entrants. At the same time, rapid prototyping among module producers could be observed. At this systemic innovation phase, coordination is difficult among different module producers to achieve compatibility across components. This difficulty is increased especially upon various constraints imposed by technical requirements. In this stage of production, vertical integration has the advantage to coordinate the production of modules and their integration by decreasing the integration cost.

Modularity is a double-edged sword for incumbent firms. Systemic innovation could be observed within the externally compatible components (Langlois and Robertson, 1992). These compatible new products can contribute to the market success of a product. In the long run, entry of firms producing compatible products within the whole system could also undermine the market dominance of product of incumbent firms (Baldwin and Clark, 1997). In the case of a fast technological change, modularity offers an advantage of speed for new entrants while leaving the incumbents behind (Brusoni and Fontana, 2005). An incumbent can also resist to modularity by opposing to the development of a standardized interface which would incite the entry of many new module producers (Schilling, 2000). However after standardization, the place of the incumbent is not only threatened by new module producers but also by an eventual introduction of an architectural innovation. In this case, modules and interfaces are unchanged but the architectural innovator would take the lead and leave the incumbent behind (Henderson and Clark, 1990).

Modular architecture leads to independent innovation activities to be carried out without the need to change the whole system. Conway (1968) argues that any organization which develop a modular software system should also have a working communication system between separate module and interface designers i.e. the software system structure is the reflection of the social structure of developing organization(s). This is also known as the Conway's law. In the same line, Sanchez and Mahoney (1996) argue that firms producing modular products are also urged to switch to the organizational modularity. This fact creates a pressure on firms to extend their knowledge beyond what is required for their production and forces them to make sure that modules are compatible with the architecture (Baldwin and Clark, 1997). It is argued that these greater knowledge boundaries help firms to cope with the technological complexity and deal with the product level interdependence. On the other hand, it is claimed that an increasing modularity involves an increasing specialization at the firm level (Langlois and Robertson, 1992).

4.1.3 Modularity in software products

Software is inherently a complex product and there is no silver bullet to overcome its complexity (Brooks, 1987). However, the decomposition of a software project into modules is one of the solutions to develop complex systems (Parnas et al., 1985). In his seminal article, Parnas (1972) sets the concepts of software modularity and object-oriented programming which contributes to the increase of software productivity. In this paper, Parnas develops the concept of information hiding. The interface of a module should reveal very little about its inner working. Modules should be compatible with defined interfaces, but their inner working is left to be designed and developed by the production site in charge of it. Information hiding and well defined interfaces give the opportunity to many programmers to develop software with closed source software libraries without interfering with their internal working.

The benefits of modular programming are: (1) managerial; multi-site development can be achieved easily with less communication between different developer groups, (2) product flexibility; change in one module will not affect the change of the system and (3) comprehensibility; the system could be studied one module at a time (Parnas, 1972). Accordingly, modular software helps to locate, debug and fix software problems easily.

In various studies it has been observed that many products are inclined towards higher modularity with time. This evolution is also traced on different software projects. It is shown that the transition of Mozilla browser from closed to FLOSS changed the structure of the project into a higher degree of modularity (MacCormack et al., 2006). Based on this example, MacCormack et al. (2006) argue that the project had been able to make an architectural innovation to attract new developers. According to Henderson and Clark (1990) it is difficult to make an architectural innovation. However, MacCormack et al. (2006) argue that, based on this single example, architectural change could be easier in software compared to physical products. The increase of modularity is also traced for the Debian GNU/Linux project[1] and it is found that the increase in modularity helps to avoid failures to spread within the system (Fortuna et al., 2011). It is argued that FLOSS tends to be more modular than proprietary software (MacCormack et al., 2006; Raymond, 1999) but this assertion can be supported with very few examples as the number of software which came out to be open is very few compared to the number of closed source software products.

Introduction of new properties into software is generally accompanied by the development of new modules. The increase of interaction between modules results in an increase of complexity of the system. However, according to Schilling (2000), not all systems move toward an increasing modularity but some move toward increasing integration. This argument is supported with examples given from software applications that are sold bundled within an operating system.

As modularity increases it is also expected that firms knowledge boundaries widen in order to coordinate innovation and production. It is also asserted that modularity would induce specialized firms. In the next section the implications of modularity for organizational structure will be discussed. Furthermore, in the next chapter, formal analysis, using patents, will be

[1]Debian GNU/Linux is one of the most popular distribution which is composed of mainly software which are licensed with licenses that comply the Debian Free Software Guidelines, such as the GNU GPL. It is developed by volunteers.

carried out to explore the relation between modularity and organizational structure.

4.1.4 Diversification vs. specialization

The implications of technological diversity has been discussed by many scholars. According to Breschi et al. (2003) there are two competing hypothesis. The first one emphasizes that firms which are specialized in a small number of R&D fields have more innovative capability than the diversified firms. The second hypothesis claims that even though some specialization is required, firms which are more diversified have competitive edge through cross-fertilization of different but related technologies. Their results, based on patent analysis, show that knowledge-relatedness is an important factor in technological diversity of firms.

Brusoni et al. (2001) argue that multi-technology firms are competent across wider knowledge fields than what their products require. A wider knowledge base will have positive effects on the coordination of suppliers and the integration of various products for multi-technology firms. In biotechnology, the diversification of the technology base has an important effect on exploratory innovative capability (Quintana-Garcia and Benavides-Velasco, 2008). Diversity has also a positive effect on the rate of product enhancement. Among R&D intensive European firms, R&D intensity and the number of patent increases with the degree of technological diversification (Garcia-Vega, 2006). The author argues that firms which have diversified technological knowledge can reduce their risks and increase their chances to receive more spillovers. As a consequence of learning process and of the localized nature of knowledge, Breschi et al. (2003) claim that the technological diversity of a firm is linked to the knowledge-relatedness and show that, innovative activities of firms are expanding in a non-random way.

Combining the discussions on modularity on knowledge boundaries of firms and that of the knowledge diversification and specialization of firms gives the background of the research question of the current and the next chapter. To understand whether technological achievements in software technologies come from diverse or specialized firms, patent analysis will be carried out.

Patent analysis is frequently utilized in innovation studies but it should be used with caution (Griliches, 1990). It provides many opportunities to address the research questions given in this chapter. Combining simple text mining techniques and various network algorithms in a patent database, it is possible to trace the evolution of a specific technology and innovative firms within that field. In the next section various methods on patent analysis are presented, an emphasis is given to the patent citation network analysis developed by Hummon and Doreian (1989). The focus will be on the "important" patents which have shaped the development of the technology. This analysis will be applied in the next chapter to the case of video indexing software technology.

4.2 Patent analysis

Patents represent the best archive on inventions and cover practically all fields of innovation carried out in most developed countries (Jaffe and Trajtenberg, 2005). Patent documents contain information on the technological area of the invention, names of inventors, assignees of the patent, year in which the patent is filed, citations to scientific and other patents. The importance of the intellectual property rights (IPR) strategies, the rapid rise in the number of patents and the advent of the computerized patent database drive the development of the patent analysis (Griliches, 1990). The evaluation of patents is an important step in understanding technological and industrial evolution and/or designing a firm level IPR strategy (Ernst, 2003).

The assessment of a legal document such as patent is a cumbersome process. A patent is generally evaluated within a group of patents. Patent analyses are generally carried out retrospectively. On the other hand, some other methods based on text mining could determine new emerging technological fields (Kostoff, 2008). This method could help to make some predictions and support to shape IPR policies and/or R&D strategies.

In patent analysis, patents are generally described as either *important* or *valuable*. A patent is described as important when it is a reflection of a technological achievement and plays an important role in the development of a technology. In that case important patents generally determine a technological breakthrough or a consistent improvement of the technology. Valuable patents contribute financially to its owners. Financial contribution could be measured in many forms such as contribution to IPO value, Tobin's Q value, sales volume of firms and so forth. A patent could also be valuable if it is used successfully in patent litigation cases.

Automatic classification of patent documents is more in demand with the increase in the number of patents and as the patent analysis requires more sophisticated tools. There are basically two different approaches in interpreting patent data (Kostoff and Schaller, 2001). The first one is the expert driven approach which uses the assistance of the domain experts. This approach is generally time-consuming and not easy to scale with the amount of data which is easy to gather. The second approach is the computer based approach which can easily overcome the size of the collected data. However, the second approach needs generally expert assistance or complements expert driven approach.

There are many bibliometric methods which are applied in patent analysis. One of the methods used in patent analysis is the rudimentary and obvious patent features count; such as number of citations or claims which have its roots in the work of de Solla Price (1965). Other more complex text mining and citation network analysis are also used to address various research questions. These methods are applied to a group of patents which are limited by some International Patent Classification (IPC) code(s), technologies, firm(s), geographical area(s), inventor(s), product(s) or their combinations. In recent years, computer based patent analysis is improved with software including various complex algorithms. This section presents and discusses several of the patent analysis methods found in the literature.

4.2.1 Counting patent features

Simple patent counts, such as the number of patents assigned to a country, industry or firm for a defined period, are admitted to be an inaccurate measure of value indicator (Griliches, 1990; Trajtenberg, 1990). However, counting several patent features show that there are some correlations of the value and quality of the patent. Similar to the scientific papers, citation number criteria is one of the first proposed patent feature which shows a positive correlation with its economic value (Griliches, 1990). It is also argued that if a patent does not include any citation that patent could be a *pioneering* patent (Ahuja and Lampert, 2001). Another assessment of the quality of a patent is the one which has citations to scientific papers. Patents with citations to scientific papers are inventions which embrace scientific knowledge and it is found that these patents are also more cited by other patents (Gittelman and Kogut, 2003; Fleming and Sorenson, 2004). In addition to these, patent family size, the number of oppositions and renewals are also used to rate the quality of a patent (van Zeebroeck, 2011). Creative use of patent features could also give important information on a specific technology. Dahlin and Behrens (2005) use the backward and forward patent citations in order to find radical inventions which have shaped the tennis racket industry.

The value of a patent is generally measured by correlating patent features with another value which is generally one of the performance criteria of firms. With this research approach, it is found out that in biotechnology the scope of the patent portfolio measured by the number of IPC are correlated with the market value of firms (Lerner, 1994). The rise in the number of software related patents and their much discussed traits questioned their contribution to the performance of firms. The effect of software patents on pure software firm market value is investigated (Hall and MacGarvie, 2010). Authors find that there is not a clear conclusion that the number of patents found in patent portfolio of firms makes a positive contribution to its market value. On the other hand, Useche (2012) shows that each patent contribute with over a million dollar to the IPO value of the software firms in European and US market. Harhoff et al. (2003) use the same approach but utilize a data set in which the value of the patents are given by the patent holders. Authors find that backward and forward citation numbers of patents are correlated with the value of the patent given by the assignee. Moreover, patents which are in litigation procedures and patents within considerable patent families are found to be valuable. van Zeebroeck (2011) found that the five patent features; forward citations, grant decisions, families, renewals and oppositions are found to correlate with the value of patents and could also help to predict a future market for the invention. Moreover, the number of claims is also considered as a proxy for the quality of the patent (Lanjouw and Schankerman, 2004).

Patent renewal data also gives important information on the quality of patent. Patent renewal fees increases each year and if the patent owners pay this increasing sum in order to keep their patent in force, it is considered that the renewed patent is valuable as perceived by its owner. Due to this fact, patent renewals are considered as a good proxy to show the quality of a patent. A renewal based weighting of patents removes noise of the patent count as a measure of innovative activities (Lanjouw et al., 1996).

4.2.2 Text Mining

Text mining is a procedure to classify and extract information from text data. Depending on the algorithms, automatic classification or interaction of experts are the basis of this approach. Generally, a patent corpus is used to classify patents. A corpus is a set of texts which is already structured and helps to tag and categorize analyzed text. In the next chapter, n-gram, one of the basic text mining procedures, is used to classify patents. This technique is the simple measurement of n number of subsequent word occurrence in a text file. To filter unwanted results a list of stop words is also used. This filtering process requires expert help and it is also an iterative process. Text mining promises many potential discoveries with the increasing number of technical documents containing various technical concepts (Kostoff, 2008).

Text mining and automatic clustering of the patent documents follows a structured process. The first step is the collection of the patent documents relevant to the purpose of the analysis. Then the text segmentation is carried out. It is based on the well structured information such as date of filing of the patent, name of the assignees, IPC code, title, abstract, claims found in the patent document. Most of the times, these fields are already segmented if a patent database is used. Then the next step is the extraction of the co-occurring words related to the analyzed topic. The last steps are topic clustering which could be done with several well-known algorithms. Then patents are clustered according to the topics and their weights within the patent document (Tseng et al., 2007).

An example of patent analysis which uses text mining with topic modeling determines cognitive breakthrough on carbon nanotube and buckminsterfullerene (bucky balls, C60) (Kaplan and Vakili, 2012). Topic modeling uses word association and tracks the emergence of new meanings over time. It identifies patents which show technological breakthrough. Kaplan and Vakili (2012) use only patent abstracts, and experts are also consulted.

4.2.3 Citation network analysis

The scientific and technological advances are achieved by the accumulation and recombination of the previous knowledge. This is represented by the citation to the prior works in patents and in scientific papers. It is argued that the patent citation analysis is less prone to error compared to scientific citation because patent citation process is less subjective (Jaffe and Trajtenberg, 2005).

Furthermore, it is possible to trace the knowledge flows and the accumulation of knowledge on a specific technology with a citation network analysis. This flow of knowledge reveals the evolution of a technology. The definition of the technological trajectory given by Dosi (1982) forms the background of this perspective. Dosi defines technological paradigm and technological trajectory by referring to Kuhn's definition of scientific paradigms. Dosi claims that scientific and technical advances that are intertwined shape technological change which is limited with the current technological paradigm. This change is also determined by economic, institutional and social factors. According to

Dosi (1982); technological paradigm gives "*a pattern and a model of solution to a selected set of problems based on selected knowledge derived from natural science with selected materials*". As a result, technological paradigm sets the direction of technological trajectories. However, the evolution of technologies shows that while there can be lock-ins, breakthroughs could also emerge (Arthur, 1989). Discontinuities in the technological trajectory are the results of the emergence of a new technological paradigm (Dosi, 1982).

An important advantage of citation network analysis over counting patent features is its ability to give results which are traceable on a time scale. While other methods can only pinpoint breakthrough in technical or scientific field, citation network analysis involves an evolutionary approach. This method can be jointly used with text mining method to give more insights on the development of a scientific or a technical field.

An interesting example of text mining method is used with citation network analysis on scientific papers related to gallium nitride (Ga-Ni) (Shibata et al., 2008). In this work, authors cluster citation networks with keywords and obtain tightly knitted nodes. Then a time line visualization of the development of the clusters and their branching or merging could be traced. This method shows the development and evolution of some scientific topics related to the Ga-Ni materials and detects emerging scientific fields.

In this section the method developed by Hummon and Doreian (1989) which reveal technological trajectories through patent connectivity analysis is presented. This tool is firstly exploited within bibliometrics to study the development of the DNA theory in scientific journals (Hummon and Doreian, 1989). Batagelj (2003) refines the same methodology. He gives an algorithm and applies it to patents and to scientific journals to obtain the main path through which the knowledge flows. Verspagen (2007) uses the same algorithm in a heuristic approach in order to obtain a network of main paths in a patent network. This process helps to decrease the number of patents to be analyzed and give an insight to the evolution of the technology with patents representing important technological achievements in this field.

This latter method is used in the next chapter to find out technological achievements made by firms in one of the modules related to video indexing

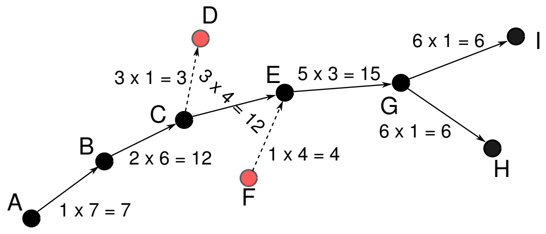


Figure 4.1: SPNP value of each edge of a representative network.

software. The result of this method is a network of patent citations, with patents which have shaped the development of the technology.

Patent connectivity analysis

Vertices in the citation network represent patents and edges represent citations between two patents. The direction of the edge is obtained from cited patents to the citing patents, revealing the direction of the knowledge flow. Vertices in the network could be a start-point, an end-point or an intermediate. A start-point is generally an old patent, which is either not citing a patent from the patent group, or not citing any patent at all. An end-point is a patent which is not cited by any other patent. An intermediate patent is a patent which is cited by and citing other patents.

In the simple network example given in Figure 4.1, vertices A and F are start-points, vertices D, H and I are end-points and vertices B, C, E and G are intermediates. The direction of the edges represents the knowledge flows. As an example; the edge B-C displays the knowledge flow from the vertex B to C, which means that the patent C is citing patent B.

A patent citation network has some distinct characteristics. First, there are not any set of edges which connects vertices in such a way that it is possible to reach to the starting vertex. Second, it is directed; all edges in a citation network represent the direction of citation and finally, it is binary due to all edges having equal weight.

Search Path Count (SPC), Search Path Link Count (SPLC) and Search Path Node Pair (SPNP) are the three ways to change a binary citation network into a weighted network. These indicators help to extract the important edges which link as many nodes as possible in the upstream and in the downstream of the knowledge flow. SPC is the simplest indicator which counts the paths between all sources and end-point vertices. The SPLC traces paths from all vertices to end-point vertices and gives the number of times an edge is found in the search path (De Nooy et al., 2005). In the next chapter the SPNP calculation method is used.

The algorithm of the SPNP calculation is given by Batagelj (2003) and a matrix based formal method is presented by Verspagen (2007). The rest of this section, the explanation of the SPNP calculation is adopted from Fontana et al. (2009). The SPNP calculates the product of the number of upstream and downstream vertices for each edge. To calculate the SPNP value of the edge C-E in the Figure 4.1, vertices found in the downstream and in the upstream of the C-E edge should be calculated. In the upstream until the edge C-E, including the vertex C, there are three vertices (A, B and C) and in the downstream starting from the edge C-E, including E, there are four vertices (E, G, I and H). As a result the SPNP value of the edge C-E is $3 \cdot 4 = 12$. This calculation is conducted for every edge, then the path with edges linking the highest SPNP value is the main path.

The second step is the identification of the main path. After the calculation of the SPNP value for each edge, the main path starts from the start-point having the highest SPNP value for its edge. In the case that there are some edges with the same SPNP values, then all edges with that value are chosen. Subsequently the same procedure is repeated from the next vertex. This procedure helps to find out the set of linked edges with the highest value from start-point(s) to end-point(s). The final result represents the knowledge flow of a citation network. Instead of the edges with the highest value, a lesser value could be set and a denser network could be obtained.

In the example given in Figure 4.1 the main paths starts from A and F (start points), and finishes at vertices D, I and H (end-points). The knowledge flow starts from the highest valued edge which is the edge A-B in our case. The edge F-E is not considered because the SPNP value of that edge is lower than the edge A-B. Then the process continues from the vertex B until the end-point which is on the highest valued edge.

Verspagen (2007) has contributed to this methodology by filtering the citation network for different time periods to determine the technological trajectory of the fuel cell technology. In this heuristic approach, a sub-network from the main citation network is obtained by extracting patents for a time span starting from the oldest one. Then subsequently the same time span is increased for each new set while keeping the first patent the same. In order to illustrate this methodology; a set of patents granted from 1965 to 2010 is obtained. Then group of patents which are granted between 1965 to 1970, 1965 to 1975, 1965 to 1980, and so forth are set. For each subset the SPNP calculations is conducted, then by adding up all main paths *temporal network of main paths* are obtained. This new network which is less dense than the whole network shows the development of the main paths. If the sub-networks are obtained with a smaller time span then there is a high probability of a temporal network of main paths with a higher number of vertices.

Verspagen (2007) uses this methodology to show the technological trajectory of fuel cells. There are other studies which show knowledge flows in various technological fields such as the treatment of coronary artery disease (Mina et al., 2007), data communication standards (Fontana et al., 2009), artificial disc (Barberá-Tomás et al., 2010), telecommunications switching industry (Martinelli, 2011), semiconductor miniaturization (Epicoco, 2013) and mobile communication standards (Bekkers and Martinelli, 2012). Patent connectivity analysis differs largely from other patent analysis due to its inclusion of time perspective.

4.2.4 Limits of the patent analysis

Patent citations provide many advantages and are used throughout many researches. However, patent citation analysis has some downsides. One of the fundamental one is its bias for the older patent in counting the number of citations. Moreover, it is shown that patent examiners and firm level patenting practices have an influence on citations and this influence could differ according to the technology concerned (Alcácer et al., 2009). Furthermore, it is argued that the patent citation is a noisy proxy for determining knowledge flow (Gomes-Casseres et al., 2006).

It is also reported that the patent drafting style could be deliberately deficient (Stevnsborg and de La Potterie, 2007). A patent database which includes all patent information is expensive and in some cases it is difficult to conduct statistical analysis. On the other hand, patent databases like PATSTAT which allow an easy statistical analysis and which is also affordable contain only patent title and abstract. A patent analysis which uses text mining on patent title and abstract could give the impression that the result could be flawed due to missing patent information such as claims. However, Fall et al. (2003) show that the use of the first 300 words of the patent abstract, claims and description gives a better result in patent classification than the use of the whole patent document.

Related to the software patents, the quality of patents filed by non-practicing entities (NPE, or patent trolls) are generally considered to be low quality which generally do not reflect any technological development.

4.3 Summary

This chapter gives a review on modularity and patent analysis which constitute a background for the research questions addressed in the next chapter. The aim of these two chapters is to understand the knowledge boundaries of firms which obtained important patents in a software domain for the development of technology.

Research on modularity started mainly in the field of complexity. Software is one of the first fields in which research on modularity was conducted, yet, many other fields were also analyzed. Modularity is used to overcome the increasing knowledge complexity and the speed of technological change. It affects product design as well as organizational structure. One of the discussions within the modularity research is its relation to the organizational structure in terms of knowledge boundaries of firms.

The research questions which will be answered in the next section are; 1) Do technological achievements in a highly modular software technology originate from firms which are diverse or specialized? 2) What are the knowledge boundaries of firms contributing to the development of a software technology?

To answer these questions, patent citation analysis will be carried out. Results obtained from the patent citation analysis decrease the number of patents to be analyzed and give important patents and firms which have shaped the development of the analyzed technology. Moreover, this method provides the development of the technology in a chronological order. Subsequently, technological diversity of firms is obtained.

The analysis will be conducted on video indexing, a highly modular technology aimed to extract information automatically from video files to make content easier for searching. Main modules of video indexing are optical character recognition (OCR), sound and speech recognition and image analysis. Each module is also used separately in different products showing the economies of scope within this field.

Chapter 5

Effects of modularity on organizational structure in the software industry

In the previous chapter, modularity is presented with respect to its effect on product (Ulrich, 1995) and on institutions, interfirm and intra-firm organizations (Garud and Kumaraswamy, 1995). Its effect on knowledge boundaries of firms is also discussed. Moreover, effects of modularity which change according to the development phase of the technology are also taken into account (Langlois and Robertson, 1992). In the previous chapter, these issues as well as patent analysis were presented to give the theoretical and methodological background. Although there are some studies on modularity in software industry, the literature is not well developed about the relation between modularity and knowledge bases of firms active in software industry. This chapter aims to fill this gap.

Research questions of this chapter are as follows; 1) do technological achievements in a highly modular software technology originate from firms which are diverse or which are specialized? 2) What are the knowledge boundaries of firms contributing to the development of a software technology?

In this chapter, the above research questions are dealt with the case of video indexing technology. This software technology is chosen because it is very modular and requires the technological combination of different fields of software. Its main modules which are optical character recognition (OCR), speech/sound analysis and image analysis are already in use in various sectors and in different products. This fact represents a good example of software in economies of scope which is investigated in Chapter 3.

Patent connectivity analysis is the main methodology used in this chapter. This method reduces the number of patents to analyze, shows the technological trajectories which are adopted through years and provides information on patents and the names of companies which have made technological achievements in a distinct technological area (Verspagen, 2007). This method is used to answer the first research question by providing firm names. To answer the second question, knowledge diversification of firms which have contributed to the development of the industry is calculated. In the last step results are compared to the average of the industry. Patent connectivity analyses start from 1970s and onwards. This work is important to understand whether specialized firms versus diverse firms are better off in defining the development of specific software technologies, in our case the video indexing and its modules.

Video indexing is a key software technology for the automatic information extraction from video files and their classification. Search in video and photo on Internet still depends largely on tags and keywords given by users. Video indexing technology, which is still a developing technology, has the potential to become very important in web search and has the power to widen the web search market (Bughin et al., 2011). Moreover, this technology has the potential to be used in areas such as video on demand TVs, video archives, and video surveillance. Video indexing technology helps to find pattern, image, sound or text within a video database and correlate these information with other video for an accurate description of the content. This technology is also very important for the surveillance industry. It can help to identify, track and find objects as well as people from images obtained from surveillance cameras. Video indexing requires textual, sound/speech and image analysis. These three technologies are more mature compared to video indexing and are used separately in different products. These three technologies are the main modules of the video indexing.

This chapter continues with an overview of the video indexing technology. This section presents the technological development of video indexing and its three main technological modules. Section 5.2 presents the theoretical background and the research plan of this chapter. In Section 5.3 methodology used in patent analysis is briefly discussed. Section 5.4 presents how the data used is obtained. Results obtained from these methods are shown in Section 5.5 and in Section 5.6 results are discussed, and the final section concludes.

5.1 An overview of video indexing

Video indexing enables to "understand" the content of a video file and generates a semantic meaning for specific time frame within a file. The manual construction of an index on digital video database is a time consuming activity which was once carried out by archivists (Snoek and Worring, 2005). However, as the number of video data increases, an automatic, computerized analysis to annotate and classify videos became a necessity.

Video indexing is made of numerous modules, its main modules are related to optical character recognition, audio/speech analysis and image analyses which have also different sub modules as it is conceived by the theory of modularity. These three modules are already used independently in various products and they are more mature compared to their combinatorial use in video indexing.

To find out the different modules of the video indexing, its product system architecture (Ozman, 2011) is modeled as in Figure 5.1 with the help of experts[1]. Experts provide module names which are grouped into three main modules of video indexing as presented by Snoek and Worring (2005) who develop a framework to decompose video files. In this framework, the content of a video file is decomposed into three channels which are the visual, the auditory and the textual channel. Therefore, in order to make an automatic video indexing visual analysis, audio/speech analysis and optical character

[1]Experts from Alcatel-Lucent Bell Labs, Nozay, France and researchers from Institut-Mines Telecom SudParis are addressed within the project Ubimedia.

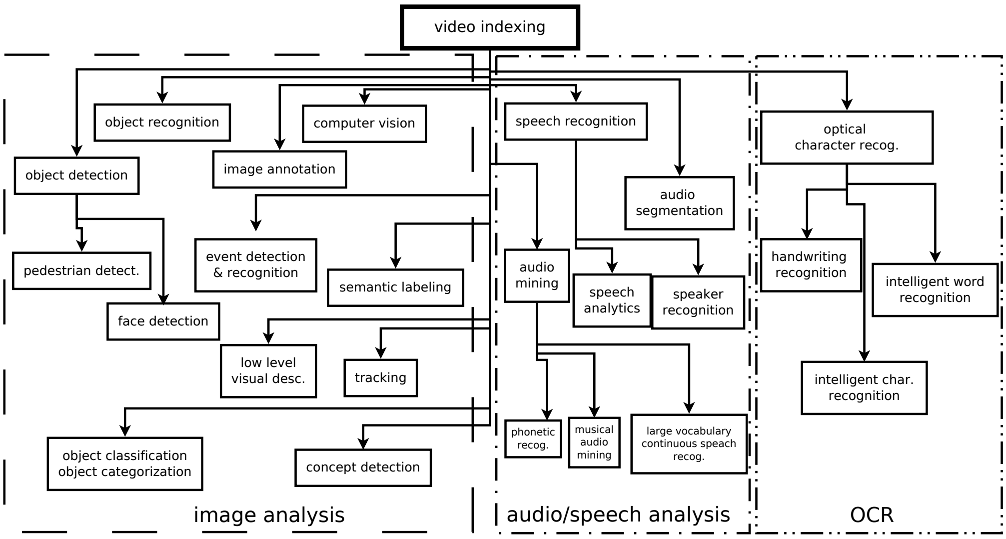


Figure 5.1: Product architecture of the video indexing.

recognition technologies are required. In this section, video indexing technology and how its three modules work for information retrieval from video files are presented. Historical development prior to 1970 of the modules are shortly presented with the results.

The general approach to extract and to index content of video files is to use the unimodal approach. In this approach, image analysis, sound and/or speech analysis and/or textual analysis are conducted separately. However, the multimodal analysis approach uses various information obtained from video files in parallel to attain more precise results. In the multimodal analysis, image frames, sound tracks, text obtained from the image frames and spoken words which could be fetched from the audio track are used in combination to increase the precision of the analysis. In some cases, information obtained is cross checked with information obtained from different video (Wang et al., 2000). Other techniques, which are used in combination, are detection of scene change by visual and audio analysis, camera motion, object detection, pattern recognition, object tracking, event detection, event recognition, etc. One of the technical hurdles in video indexing is the use of all the information obtained through various analysis from a video database, and creating a form such that their combination gives a correct semantic information. Employing different analysis techniques on large video databases in combination with real time video image flow such as in a city wide surveillance system is an important engineering problem. The advantage of a multimodal analysis is the interaction of modules which increases the accuracy of the results. Addition of new modules which are in charge of different analysis tasks would increase the precision of the video indexing. Other than some experimental, proof-of-concept software, it is reported that there are some commercial products which use multi-modal analysis (Snoek and Worring, 2005).

An early example of multimodal approach is a system which associates names and faces in news videos without prior face-name association set (Satoh et al., 1999). Schematic representation of this technique is given in Figure 5.2. This system aims to find images of a searched name within a news video database without prior training. The system uses several sources of information available in the video. Names are extracted from transcript and video-captions. The transcript of the video is generated by speech recognition analysis. Then, an in-depth semantic analysis is carried out to retrieve names from the transcript text. Facial images are also extracted from the video through image analysis. In news videos, names generally appear in video caption. Names appearing in the video captions are extracted through character recognition. All these information obtained from the videos are recorded with their time-code[2]. Then a correspondence is created between information obtained from various sources. After gaining as much information as possible from the video content, co-occurrence factor is calculated to match names with faces. However, the example given by Satoh et al. (1999) is only one example of matching names and images from news videos. There are various types of videos, such as those obtained from surveillance cameras, which lack easy processing information.

The result of the extracted information of a video content is accessible through a multimedia content descriptor. This descriptor contains time-code

[2]Time-code is a label used in each video frames to provide a time reference to edit video and to synchronize image and sound.

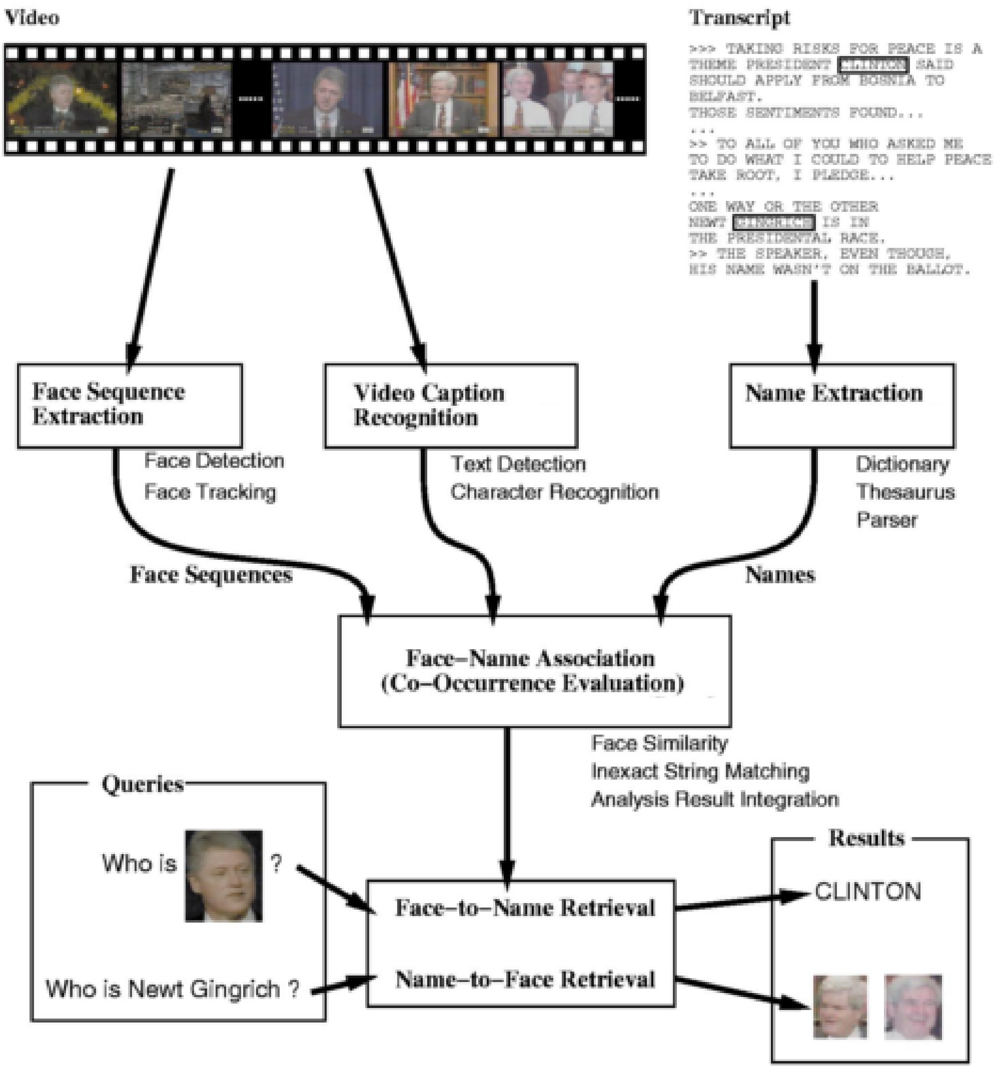


Figure 5.2: An example of multi-modal video indexing (Satoh et al., 1999).

and semantic description for particular events within the video. The open content description standard MPEG-7 was first released in 2002. The content descriptor file which is apart from the audiovisual content is generated to be used by search engines to give pertinent answer to the searched item(s).

Modular architecture of the video indexing stimulates product differentiation. Through different reconfigurations of modules, it is possible to design products for different markets. For example; the end product could be used by content providers which would index their multimedia content. A trimmed version of the video indexing product could also be used in consumer electronics market by indexing the ever growing private, amateur video content.

5.2 Theoretical background, research plan

The aim of this chapter is to find whether firms, which have shaped the development of software technology through technological achievements, are technologically diverse or specialized, and how these firms relate to the general development of firms which contribute to this technology. To answer these research questions, video indexing is chosen as the investigated case. In the previous section it is shown that video indexing is a modular technology which is represented in Figure 5.1. A longer theoretical background of this chapter is given in the previous one.

Modularity is found to be related to different concepts. Modularity helps to overcome the increasing complexity of knowledge which underlies products and cope with the speed of the technological change. Modularity has its effect on product (Ulrich, 1995) and also on interfirm and intra-firm organizations (Garud and Kumaraswamy, 1995; Sanchez and Mahoney, 1996). According to Langlois and Robertson (1992), modularity results with vertical and horizontal disintegration of firms and eventually results with specialized firms. However, Schilling (2000) argues that not all systems evolve into modularity and some could move towards increasing integration.

Firm profile within the modular production could change according to the development phase of the technology. In the early stage of the development of a technology new entrants can contribute to the success of a product with their compatible products. These new entrants can also threaten the position of incumbents and can surpass them (Baldwin and Clark, 1997). Incumbents can also resist to the standardization of interface which would increase the number of new entrants (Schilling, 2000).

In order to answer the research questions, patent analysis is conducted. In the first step, patent groups are formed for each of the modules of the video indexing. Then patents which represent the technical achievements in each module are obtained by the citation network analysis (Verspagen, 2007). Results are given in the form of knowledge flow maps. The advantage of this method is that it gives technical achievements within a time-line providing also the development branches of the technology.

In the second step knowledge boundaries of firms which made technological achievements and the average of the industry is analyzed. Innovative capacity of firms according to their technological diversity is subject to debate (Breschi et al., 2003). In one side, it is argued that firms, by cross-breading of different technologies, can achieve higher innovative capabilities. On the other side, it is also argued that firms which are specialized in few R&D fields can have higher innovative capacity than the diversified ones.

5.3 Methodology

The main method used in this research is based on patent analysis. Patent analysis provides important information on the development of a technology within a time frame. In the previous chapter, in Section 4.2, various patent analysis methods are presented. In this chapter, to depict technological achievements, which have shaped the development of one of the modules of video indexing, patent connectivity analysis developed by Hummon and Doreian (1989) and refined by Verspagen (2007) is used. This method is presented in Section 4.2.3 and in this chapter, Search Path Node Pair (SPNP) algorithm is used. Through this method, patents which represent considerable technological achievements are obtained for video indexing and its modules. Then in the second part, knowledge diversity of firms, which have made technological achievements, are analyzed. The method used to obtain the technological diversity index is given next.

Technological diversity index

Technological diversity of a firm is deduced by the number of patents obtained in distinct technology groups. In order to classify patents in different technologies, IPC (International Patent Classification) codes are used. This classification is based on 35 different technological fields and given in Table 5.1. This table is also known as the WIPO IPC-Technology Concordance Table[3] (Schmoch, 2008).

Table 5.1: Technological fields.

1	Electrical machinery, energy	19	Basic materials chemistry
2	Audio-visual technology	20	Materials, metallurgy
3	Telecommunications	21	Surface technology, coating
4	Digital communication	22	Micro-structural and nano-tech.
5	Basic communication processes	23	Chemical engineering
6	Computer technology	24	Environmental technology
7	IT methods for management	25	Handling
8	Semiconductors	26	Machine tools
9	Optics	27	Engines, pumps, turbines
10	Measurement	28	Textile and paper machines
11	Analysis of biological materials	29	Other special machines
12	Control	30	Thermal processes and apparatus
13	Medical technology	31	Mechanical elements
14	Organic fine chemistry	32	Transport
15	Biotechnology	33	Furniture, games
16	Pharmaceuticals	34	Other consumer goods
17	Macromolecular chemistry, polymers	35	Civil engineering
18	Food chemistry		

The technological diversity is defined with the Herfindahl index given in equation 5.1.

$$H = \sum_{i=1}^{35} (\frac{n}{N})^2 \tag{5.1}$$

[3] http://www.wipo.int/ipstats/en/statistics/patents

i represents one of the 35 technological fields given by the WIPO IPC-Technology Concordance Table. n is the total number of patent within a particular technological field and N is the total number of patents. In this chapter, patents for the period of the last five years are used to retrieve the technological diversity of a firm. The value of H (Herfindahl index) ranges between 0 and 1. The diversity index is $1 - H$, the greater the value of the index, the greater the diversity.

Follow-up is an example of the calculation of the diversity index of VideoMining Corp. for the year 2004. The technological diversity of VideoMining Corp. is calculated by taking into account all patents of the company filed for the last five years. N, is the total number of patents filed by the same company between the start of the year 2000 and the end of 2004. n is the number of patents which is found in one of the 35 technological field given in the IPC-Concordance table. The diversity index of the firm Video Mining for the year 2004 is obtained by using equation 5.1 and the result is subtracted from one.

In order to start the patent analysis, patent groups should be formed according to the modules OCR, speech/sound analysis and image analysis. In the next section the process of data collection is presented. Subsequently, from these patent groups, first, patent citation networks are presented then by using the results of the citation networks, diversification levels of firms and industries are given.

5.4 Data

The aim of this analysis is to carry out a main path analysis for the three modules of video indexing and the video indexing itself. For this purpose, patent groups have to be created for each module and their intersections are aimed to be minimized.

Keyword based search with stop words and co-word analysis are carried out on patent databases in order to create all modules related patents. This method helps to refine the patent group and very often it is used in an iterative way. This method is required because software technologies do not have any distinct International Patent Classification (IPC) code. With this method a coherent core patent group is created.

To find out the necessary keywords which are used in identifying the initial patent set, video indexing product system architecture in Figure 5.1 is used. From this product system architecture, keywords are grouped into three main modules of video indexing. In this research, EPO Worldwide Patent Statistical Database[4] is used. Keyword based search is applied only on titles and abstracts of patents granted by USPTO. PATSTAT does not contain any other textual field like the patent claims which might give more precise results for keyword searches. On the other hand, a patent database which allows full text search might also cause to obtain false positives by increasing the number of patent documents containing one of the keywords. Further keyword determinations are done iteratively by using n-gram analysis several times on the text generated by patent titles and abstracts. The n-gram analysis helps to obtain keywords which might be ignored during the discussions with experts. Having acquired keywords and patents, some of the patents, which change the composition of the patent group, are discarded. In image analysis, experts suggest not to use some of the patents containing words such as "vehicle" and "car" because the end results are very much related to image processing used in driverless cars. In audio analysis, several patents related to video analysis are found in the result set. In order to limit the analysis solely on one technology, i.e. OCR, image analysis etc., some stop words[5] are used. In defining patents representing speech and audio analysis patents having the "video" keyword is not included in this group. In image analysis patents containing keywords such as vehicle, car, video, speech, audio and sound are omitted. However, in defining video indexing experts insisted that all patents should contain the keyword "video" in patent titles and abstracts. All keywords used in defining the four patent groups of the modules are given in Table 5.2. This keyword based search helps to define the core patents to be populated with citations for each module.

In order to obtain a more connected patent group and also to catch

[4]PATSTAT April 2010 edition.

[5]A list of words which are used to filter the result.

Table 5.2: Keywords used for each patent groups in selecting the core group.

image analysis	**speech/audio analysis**	**character recognition**	**video indexing**
object detection	speech recognition	ocr	video*classif
object recognition	speech recognizer	optical character recognition	video index
facial detection	voice recognition	paper to computer	video retriev
facial recognition	audio segmentation	paper-to-computer	object detection AND video
face detection	audio mining	handwriting recognition	object recognition AND video
face recognition	speech analytics	intelligent character recognition	facial detection AND video
pedestrian detection	speaker recognition	intelligent word recognition	facial recognition AND video
people detection	phonetic recognition		face detection AND video
computer vision	speech to text		face recognition AND video
image annotation	speech-to-text		pedestrian detection AND video
event detection	sound pattern		people detection AND video
event recognition	silence detection		computer vision AND video
semantic labeling	speaker identification		image annotation AND video
concept detection			event detection AND video
pattern recognition			semantic labeling AND video
visual descriptor			concept detection AND video
object classification			visual descriptor AND video
object categorization			object classification AND video
			object categorization AND video
			visual descriptor AND video
			event recognition AND video
stop words			
vehicle	video		
car			
video			
speech			
audio			
sound			

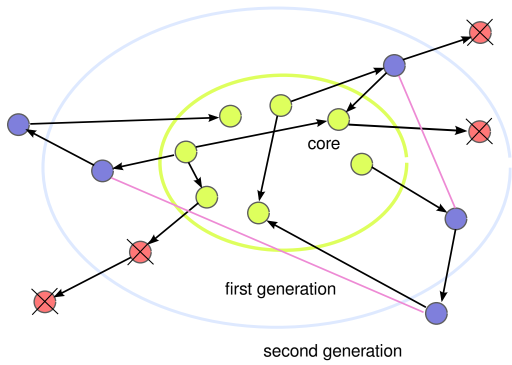


Figure 5.3: Populating the number of patents through citations (van der Heijden, 2010). Direction of arrows are from patents to the cited patents i.e. backward citation.

patents which could not be obtained by keyword search, the following procedures developed by van der Heijden (2010) is used. The schematic representation of this procedure is shown in Figure 5.3. In this procedure, the core group obtained by keyword search is populated with their cited patents. This group constitutes the first generation. Then the second generation of patents is obtained by the patents cited by the first generation patents. Then, patents which do not cite directly or indirectly the core patents within the first and second generations are eliminated. The end result is a densely connected two generations of patents citing the core patent group.

5.5 Results

In this section, first, descriptive statistics are shown then knowledge flow maps are presented. Then, the technological diversity index of firms which have made technological achievements in different modules is given. Next, the mean diversity index of firms which have contributed to the development of the modules with their patents is shown.

The distribution and the number of patents obtained for each technology is given in Table 5.3. Some patents are found in different patent groups simultaneously after the process of populating the number of patents as represented in Figure 5.3. This is caused by the usage of some technologies, such

Table 5.3: Number of patents in each group.

	core	sum of all generations	number of citation
image analysis	893	3,144	12,725
speech/audio analysis	2,777	6,781	45,602
character recognition	481	1,880	5,494
video indexing	113	240	513

as hidden Markov models, which are widely used in pattern recognition related to the modules studied. This is also another example of rich economies of scope in the knowledge base of some software technologies. The number of patents found in four patent groups are given in Table 5.4. The number of patents and their distribution by their filing year are given in Figure 5.4. The high number of patents for the period of 1995 to 2000 is due to the process of populating patents as represented in Figure 5.3. This process requires the inclusion of patents which are cited by the core patents for two generations and these patents should at least cite one of the patents in the core group. This process increases the number of some older patents giving the results shown in Figure 5.4. The top 20 highly cited patents for each module are given in the Appendix B.

Table 5.4: Number of overlapping patents.

	image	speech/audio	character recog.	video indexing
image analysis	x	479	320	131
speech/audio analysis		x	129	20
character recog.			x	2
video indexing				x

In this section, results of patent analysis of video indexing and its modules are presented in order to understand whether technological achievements in a highly modular software technology originate from diverse firms or specialized firms. Knowledge flow maps give the patents which represent technological achievements. From these maps firm names are obtained. Accordingly, it is assumed that these firms made technological achievements in the year of patent filing. In the next sub-section, knowledge flow maps of all modules

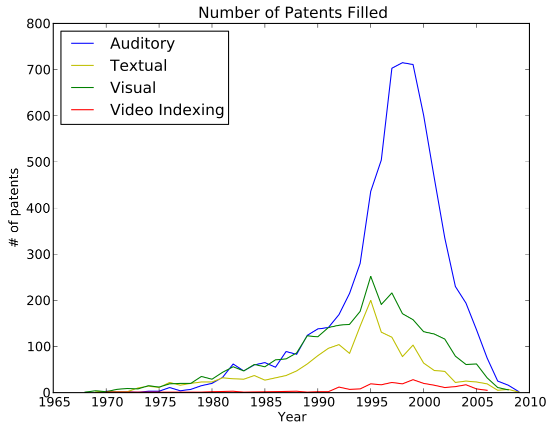


Figure 5.4: Yearly distribution of the number of filed patents.

are given.

5.5.1 Knowledge flow maps

Several branches in the network of main paths are obtained during the development phase of the three modules of the video indexing (Figures 5.5, 5.6, 5.7). Nodes, which are on the knowledge flow map, are patents which represents technological achievements in the development of the module. Moreover, these knowledge flow maps help to distinguish and categorize distinct sub-branches of the software technology. These branches do not represent different technological trajectories (Dosi, 1982) that have been adopted but are different fields on which the module develops. In each module section, technology of the module is briefly explained.

Optical character recognition

The first optical character recognition (OCR) related patent is obtained by Tauschek in 1929 in Germany and in 1933, an American inventor filed a similar patent in US. These inventions were opto-mechanical devices. During the beginning of 1960s several commercial software based OCRs appeared. The first generation of OCRs were able to read only symbols specifically designed for machine reading. The first machines which were able to recognize hand-printed characters were marketed by the end of 1960s and early 1970s for automatic postal code sorting. The characteristics of the following generation of OCR are recognition of poor print-quality characters and large hand-printed character set including Chinese characters (Mori et al., 1992). During the 1980s, several OCR accuracy contests are conducted. Some of the participant software were never commercialized. One of them which was developed from 1985 to 1995 and which was awarded in an OCR accuracy contest was released as FLOSS and endorsed by Google in 2005[6].

The network of main paths of character recognition is represented in Figure 5.5. The start of the development of the textual modality is shown with the axis denoted as "S". This starts to diverge into two different technical fields after 1983. From that patent there are two branches, one is extending to up, denoted as "U", and the other one to the down, the "D" axis on Figure 5.5. The "U" axis is the development of the text extraction and recognition from a captured image. This extraction is applied in different circumstances such as from automobile number plates and identification cards. Starting from the patent awarded to IBM in 1991 (pat.no 5343537) in the "U" axis, two clusters emerge. These two clusters of patents are both related to hand writing recognition inventions. On the other hand, patents on the "D" axis, are related to the optimization, enhancement of the optical character reader software. Firm names and patents found on the network of main paths of the OCR module are given in Table 5.5.

[6]Presentation of Svetlin Nakov, Tesseract OCR Engine at OpenFest 2009, http://www.slideshare.net/nakov/tesseract-ocr-engine-openfest-2009

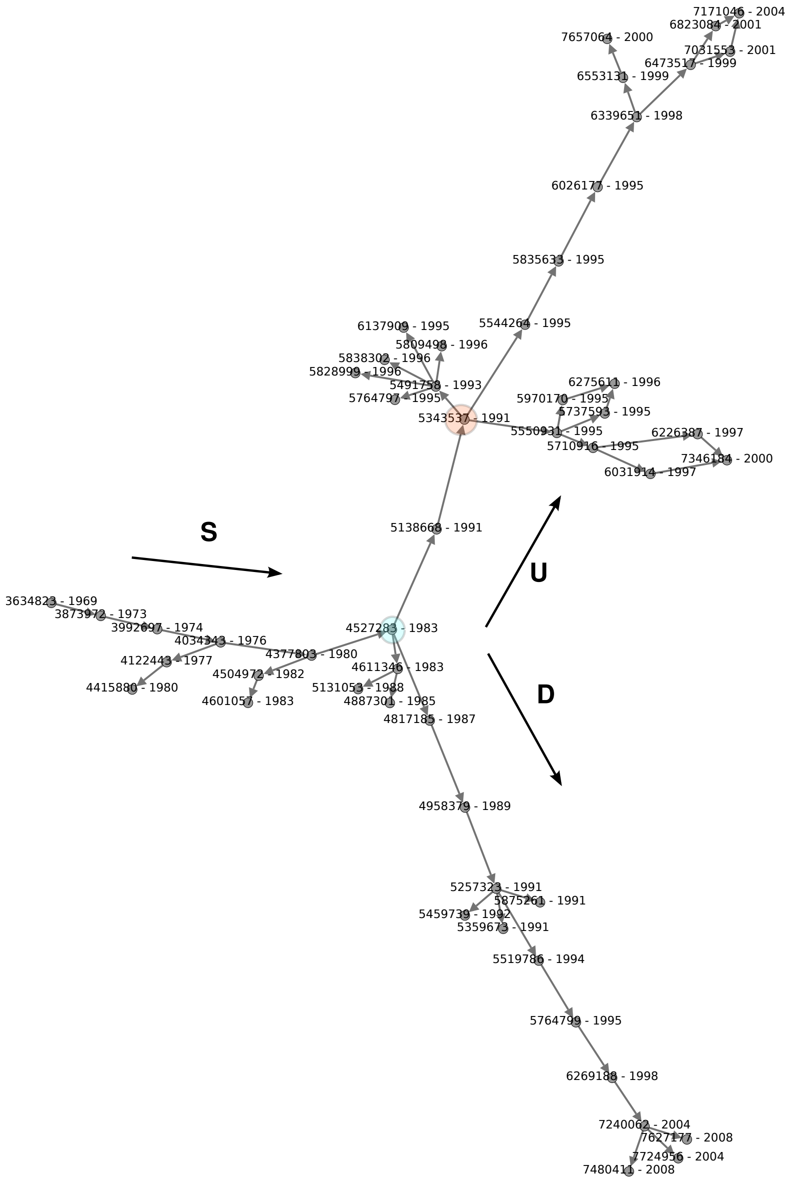


Figure 5.5: Network of main paths of the character recognition determined with SPNP. Nodes are labeled with publication number - application filing year.

Table 5.5: Firms on the network of main paths of character recognition. The location column shows the placement of patents within the related figure. “U” points out the axis stretching to up and “S” to the starting patents, and “D” to down.

Patent num	Assignees name	Pub. year	location
7480411	IBM	2008	D
7627177	IBM	2008	D
7171046	SRI International	2004	U
7240062	iArchives	2004	D
7724956	Gannon Technologies Group LLC	2004	D
6823084	SRI International	2001	U
7031553	SRI International	2001	U
7346184	Digimarc	2000	U (cluster)
7657064	Digimarc	2000	U
6473517	Siemens	1999	U
6553131	Siemens	1999	U
6269188	Canon	1998	D
6339651	Kent Ridge Digital Labs	1998	U
6031914	University of Minnesota	1997	U (cluster R)
6226387	University of Minnesota	1997	U (cluster R)
5809498	Panasonic	1996	U (cluster L)
5828999	Apple	1996	U (cluster L)
5838302	Casio	1996	U (cluster L)
6275611	Motorola	1996	U (cluster R)
5544264	IBM	1995	U
5550931	IBM	1995	U (cluster R)
5710916	Panasonic	1995	U (cluster R)
5737593	IBM	1995	U (cluster R)
5764797	Microsoft	1995	U
5764799	Research Foundation of State of State of New York	1995	D
5835633	IBM	1995	U
5970170	Kodak	1995	U (cluster)
6026177	The Hong Kong University of Science & Technology	1995	U
6137909	The USA Navy	1995	U (cluster L)
5519786	TRW Inc.	1994	D
5491758	IBM	1993	U (cluster L)
5459739	OCLC Online Computer Library Center	1992	D
5138668	Sony	1991	U
5257323	Canon	1991	D
5343537	IBM	1991	U
5359673	Xerox	1991	D
5875261	IBM	1991	D
4958379	Sumitomo	1989	D
5131053	Caere	1988	D
4817185	Sumitomo	1987	D
4887301	Dest	1985	D
4527283	Tokyo Keiki	1983	bifurcation
4601057	Omron Tateisi Electronics	1983	S
4611346	IBM	1983	S
4504972	Siemens	1982	S
4377803	IBM	1980	S
4415880	Texas Instruments	1980	S
4122443	Scan Optics	1977	S
4034343	Xerox	1976	S
3992697	Scan-Data	1974	S
3634823	International Standard Electric	1969	S

Audio and speech analysis

Using analog circuitry, Bell Labs in 1952 showed small-vocabulary recognition for spoken digit over phone. Since then important advances have been carried out regularly in automatic speech recognition but there are still technological barriers for an acceptable user experience under some conditions. Until the 1980's it was common to compare the analyzed signal to specific templates and finding the closest match. This comparison process requires high computation power. Since then, statistical models such as hidden Markov models replaced the comparison of speech to templates (O'Shaughnessy, 2008). Technical difficulties for automatic speech recognition are due to the sensitivity of the speech to the background noise, foreign accents, gender, speaking rate etc. In addition, other properties such as emotions of the person can also create different signals for the same speaker (Benzeghiba et al., 2007). Speech signals are considered to have more variability and diversity than image signals. This variance is the biggest challenge for automatic speech recognition (O'Shaughnessy, 2008).

Figure 5.6 shows the network of main paths of audio and speech analysis. The patent filed in 1988 and issued in 1991 (pat.no 5027406) the development of the audio/speech network of main paths follows two different axis; "U" and "D" respectively. This patent is about a method of speech recognition and user led training which is within the supervised learning realm. In this example the supervised learning is a function which "learns" to classify speech with the help of a human interaction. From this patent, the branch extending to the up, the "U" axis, is about speech recognition devices which are working in client-server mode. In this mode the client (PC, handheld devices etc.) with a limited capability sends voices to a server which analyzes and sends back a response. Patents on the "U" axis are focused on voice recognition in servers and interactive services. The other branch which is extending down, the "D" axis, is mainly dealing with methods on training of speech recognition systems used for speaker identification, voice activated devices and speech-based authentication. Table 5.6 shows the firm names and patents found on the network of main paths of the speech/audio analysis.

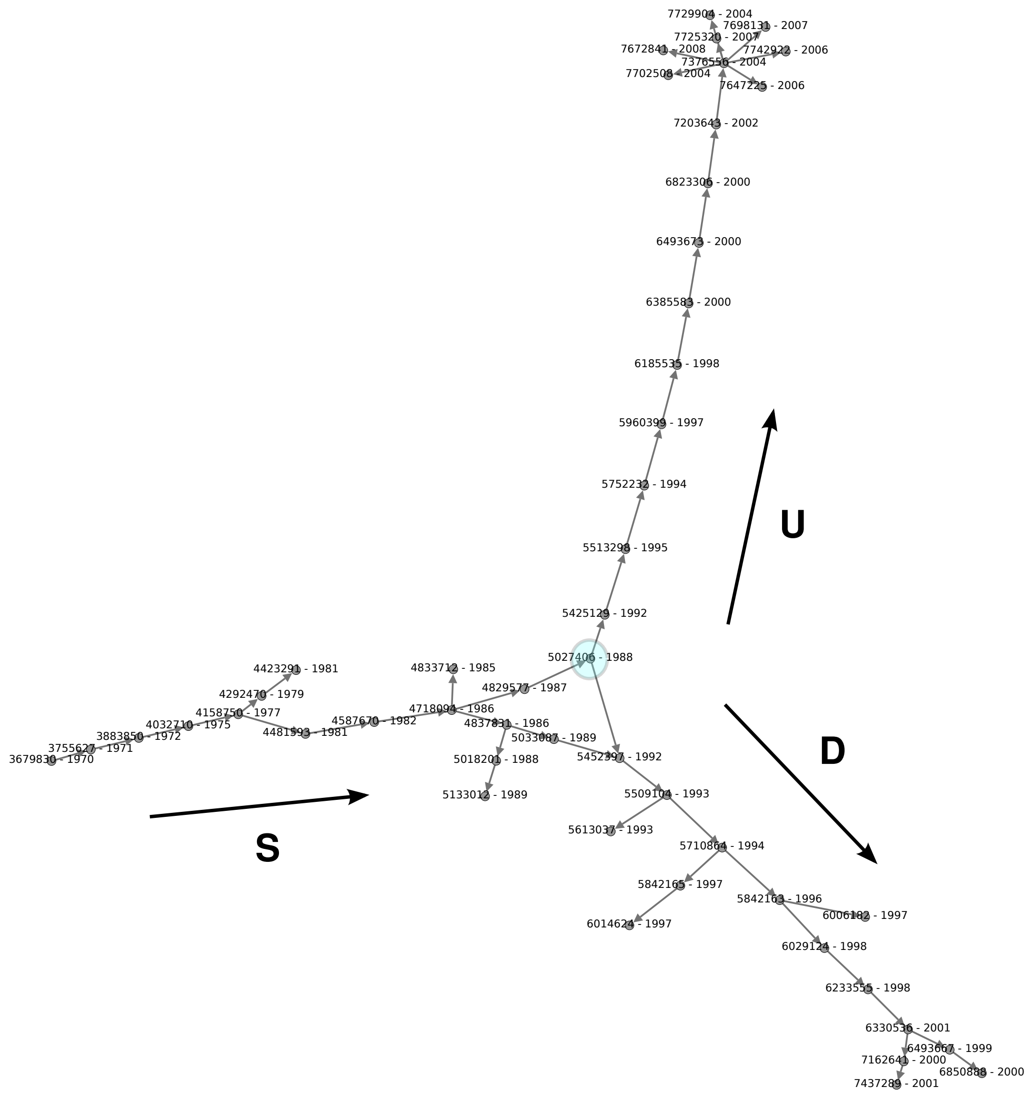


Figure 5.6: Network of main paths of the speech/audio analysis determined with SPNP. Nodes are labeled with publication number - application filing year.

Table 5.6: Firms on the network of main paths of auditory. The location column shows the placement of patents within the related figure. U points out the up and D to down.

Patent num	Assignees name	Pub. year	location
7672841	Phoenix Solutions	2008	U
7698131	Phoenix Solutions	2007	U
7725320	Phoenix Solutions	2007	U
7647225	Phoenix Solutions	2006	U
7376556	Phoenix Solutions	2004	U
7702508	Phoenix Solutions	2004	U
7729904	Phoenix Solutions	2004	U
7203643	Qualcomm	2002	U
6330536	AT& T .	2001	D
7437289	IBM	2001	D
6385583	Motorola	2000	U
6493673	Motorola	2000	U
6823306	Telesector Resources Group	2000	U
6850888	IBM	2000	D
7162641	IBM	2000	D
6493667	IBM	1999	D
6029124	Dragon Systems	1998	D
6185535	Ericsson	1998	U
6233555	AT& T	1998	D
5842165	Nynex Science & Technology	1997	D
5960399	GTE Internetworking	1997	U
6006182	Northern Telecom	1997	D
6014624	NYNEX Science and Technology	1997	D
5842163	SRI International	1996	D
5513298	IBM	1995	U
5710864	Lucent Technologies	1994	D
5752232	Lucent Technologies	1994	U
5509104	AT& T Corp.	1993	D
5613037	Lucent Technologies	1993	D
5425129	IBM	1992	U
5452397	Texas Instruments	1992	D
5033087	IBM	1989	S
5133012	Toshiba	1989	S
5018201	IBM	1988	S
5027406	Dragon Systems	1988	S
4829577	IBM	1987	S
4718094	IBM	1986	S
4837831	Dragon Systems	1986	S
4833712	IBM	1985	S
4587670	AT& T Bell Laboratories	1982	S
4423291	Siemens	1981	S
4481593	Exxon	1981	S
4292470	Interstate Electronics	1979	S
4158750	Nippon Electric	1977	S
4032710	Threshold Technology	1975	S
3883850	Threshold Technology	1972	S
3755627	US Navy	1971	S

Image analysis

Image analysis is a very broad technology which found its usage in many different areas. It would be an audacious move to try to give a short review on image analysis which requires many different techniques and approaches to be used together in order to acquire a maximum information from an image. One of the very widely used techniques, which is also used in video indexing for individual frames, is the image segmentation. It is the process to partition an image into parts which are simpler to analyze. This requires locating objects and boundaries within the image. After the image segmentation, objects could be "recognized" or labeled and classified for the search process.

The network of main paths of the image analysis is given in Figure 5.7. The axis towards up, the "U" axis, contains patents which are related to pattern recognition systems in image but also to speech. The patent on the trifurcation is the first in the series of image recognition systems using neural network systems found on the "U1" axis. However, on the same axis, patents filed after 1998 and issued after 2000 are less generic and more concentrated on face recognition systems. Last patents filed after 2003 and issued in 2009 are all related to classification systems for consumer digital images and face detection and recognition. The "U2" axis is about pattern classification which extends until 1996 and the "D" axis extends until 1986. Table 5.7 gives the name of firms and patents which appear on the network of main paths of image analysis.

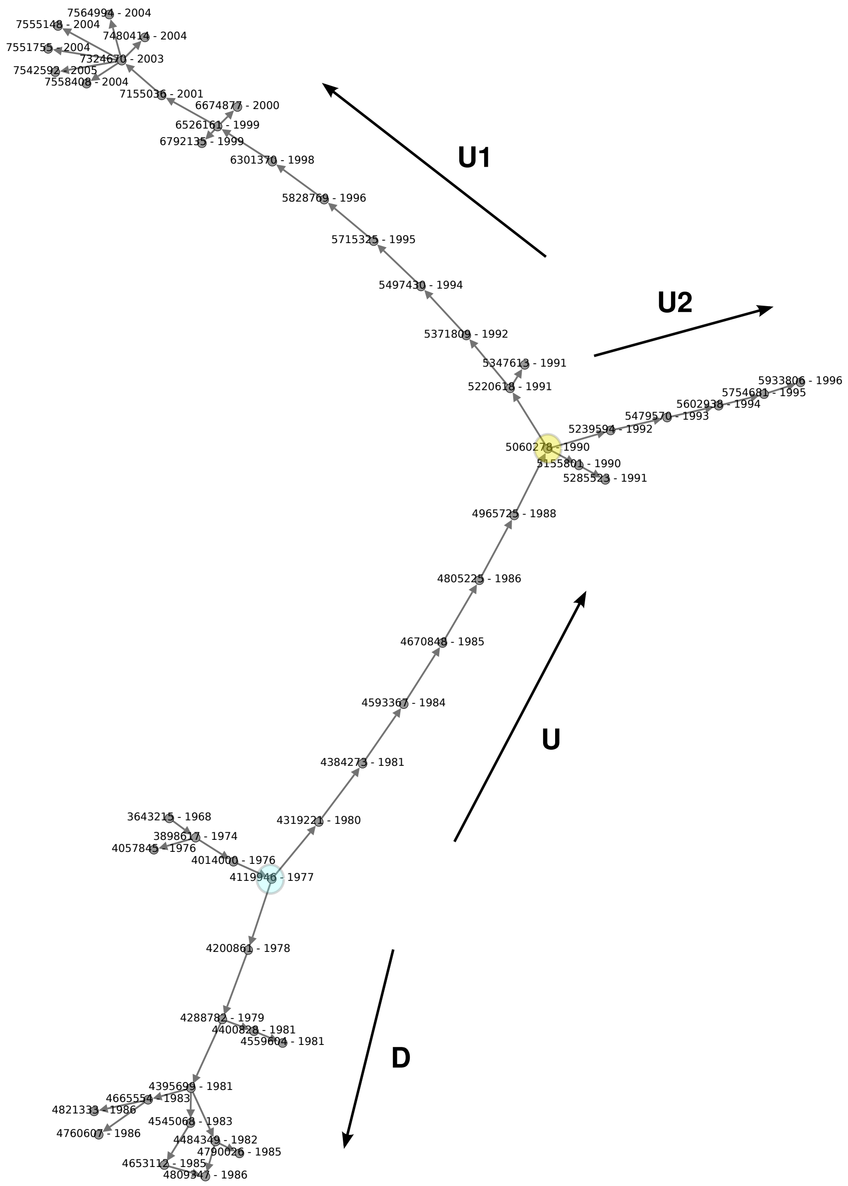


Figure 5.7: Network of main paths of the image analysis determined with SPNP. Nodes are labeled with publication number - application filing year.

Table 5.7: Firms on the network of main paths of the image analysis.

Patent num	Assignees name	Pub. year	location
7542592	Siemens	2005	U1
7480414	IBM	2004	U1
7551755	FotoNation Vision Limited	2004	U1
7555148	FotoNation Vision Limited	2004	U1
7558408	FotoNation Vision Limited	2004	U1
7564994	FotoNation Vision Limited	2004	U1
7324670	Toshiba	2003	U1
7155036	Sony	2001	U1
6674877	Microsoft	2000	U1
6526161	Philips	1999	U1
6792135	Microsoft	1999	U1
6301370	Eyematic Interfaces	1998	U1
5828769	Autodesk	1996	U1
5933806	Philips	1996	U2
5715325	Princeton University,Siemens	1995	U1
5754681	ATR Interpreting Telecommunications Research Laboratories	1995	U2
5497430	Physical Optics Corporation	1994	U1
5602938	NTT	1994	U2
5479570	Matsushita	1993	U2
5239594	Mitsubishi	1992	U2
5371809	Desieno; Duane D.	1992	U1
5220618	Philips	1991	U1
5285523	Nissan	1991	U3
5347613	Samsung	1991	U1
5060278	Ricoh	1990	trifurcation
5155801	Hughes Aircraft	1990	U3
4965725	Nueromedical Systems	1988	U
4760607	Machine Vision International	1986	D
4805225	State University of New York	1986	U
4809347	Hughes Aircraft	1986	D
4821333	Environmental Research Inst. of Michigan	1986	D
4653112	University of Connecticut	1985	D
4670848	Standard Systems Corporation	1985	U
4790026	USA NASA	1985	D
4593367	ITT	1984	U
4545068	Tokyo Shibaura	1983	D
4665554	Machine Vision International	1983	D
4484349	Environmental Research Institute of Michigan	1982	D
4384273	Bell Telephone Laboratories	1981	U
4395699	Environmental Research Institute of Michigan	1981	D
4400828	Bell Telephone Laboratories	1981	D
4559604	Hitachi	1981	D
4319221	Nippon Electric	1980	U
4288782	Compression Labs	1979	D
4200861	View Engineering	1978	D
4119946	National Research Development	1977	bifurcation
4014000	Hitachi	1976	S
4057845	Hitachi	1976	S
3898617	Hitachi	1974	S
3643215	Electric & Musical Industries	1968	S

Video Indexing

The network of main paths of video indexing, given in Figure 5.8, is represented by a single axis. Patents obtained from the network of main paths of video indexing are all related to information retrieval from a video file. The first patents on the main path of video indexing are about television systems. Some topics in this patent batch are related to generating formatted information on videos for electronic publishing, individual recognition on videos, object recognition, facial sensing and extraction of facial biometric data from image. The last patent is about scene analysis and detecting moving objects which are occluded in the background image. From the assignee name (Honda Motor Co.Ltd.) it can be deduced that the two patents which are related to the detection of moving objects from a video stream are related to the driverless cars. Firms which have made technological achievements in video indexing and their patents which appear on the network of main paths are given in Table 5.8.

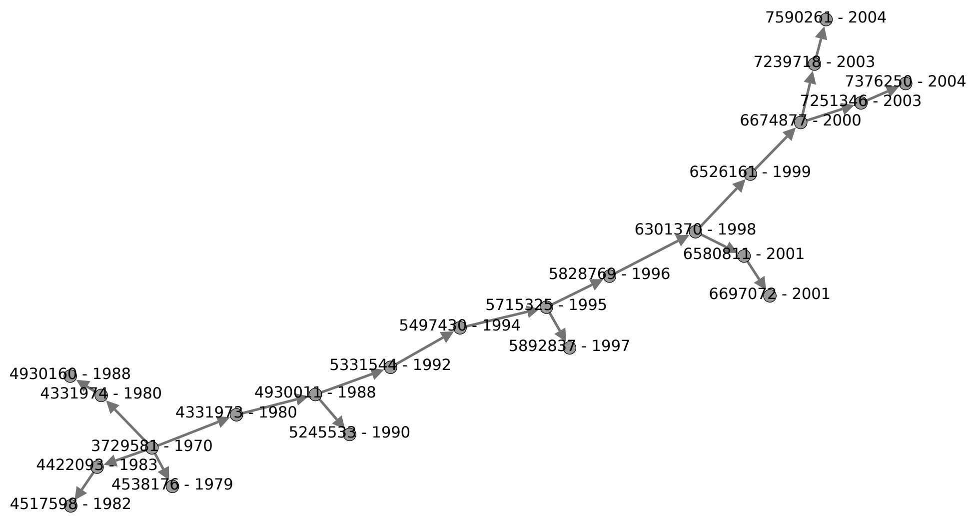


Figure 5.8: Network of main paths of the video indexing determined with SPNP. Nodes are labeled with publication number - application filing year.

Table 5.8: Firms on the network of main paths of the video indexing.

Patent num	Assignees name	Pub. year
7376250	Honda Motor Co., Ltd.	2004
7590261	VideoMining Corporation	2004
7239718	Electronics and Telecommunications Research Institute	2003
7251346	Honda Motor Co., Ltd.	2003
6580811	Eyematic Interfaces, Inc.	2001
6697072	Intel Corporation	2001
6674877	Microsoft Corporation	2000
6526161	Koninklijke Philips Electronics N.V.	1999
6301370	Eyematic Interfaces, Inc.	1998
5892837	Eastman Kodak Company	1997
5828769	Autodesk, Inc.	1996
5715325	The Trustees of Princeton University,Siemens Corporate Research, Inc.	1995
5497430	Physical Optics Corporation	1994
5331544	A. C. Nielsen Company	1992
5245533	A. C. Nielsen Company	1990
4930011	A. C. Nielsen Company	1988
4930160	Vogel; Peter S.	1988
4422093	Eeco Incorporated	1983
4517598	Van Valkenburg; George	1982
4331973	Iri, Inc.	1980
4331974	Iri, Inc.	1980
4538176	Nippon Telegraph & Telephone Public Corporation,Hitachi, Ltd.	1979
3879133	Compaagnie Electro-Mecanique	1973
3814521	Hoffmann La Roche Inc.	1972
3729581	Display Sys. Corp.	1970

5.5.2 Firm diversity

In the previous section, firms which made technological achievements in video indexing and its modules are found. In this section, firm diversity calculations of these firms are carried out to find out whether firms which have made some technological achievements are technologically diverse or specialized firms. Moreover, to compare diversity index of these firms to the average value of firms which have contributed to the development of the modules, industry level diversity index is calculated.

Results of the two diversity index calculations are given in Figure 5.9. The method to calculate the technological diversity of firms is given in Section 5.3. Herfindahl index is used to calculate the technological diversity for the four modules for each year. Each row in Figure 5.9 represent one of the four modules analyzed. The left hand side of the Figure 5.9 gives the diversity index of firms which appeared on the knowledge flow maps. On the right hand side of the Figure 5.9, the diversity index of all firms found in the analyzed patent sets is calculated and the mean value for each year is given.

To calculate the value of a diversity index of a firm for a specific year, patents which are filed for the previous five years are taken into account. The diversity index is calculated by adding up all patents filed by the firm for the last five years then the methods given in Section 5.3 and the equation 5.1 is used. Results of this analysis is labeled as “Firms on SPNP” on the trellis graph given in the left hand side of the Figure 5.9.

The arithmetical mean of diversity indexing of firms of one of the modules for each year is calculated first, by finding all firms which have given at least one patent into one of the four modules. Then, the diversity index for each firm is calculated and the mean is obtained for that year. This procedure is repeated for each year starting from 1970 until 2010. The graph related to this analysis is given in the right hand side of the Figure 5.9 under the name “Diversity Mean”.

The idea of the trellis graph is to position diversity index of firms which made a technological achievement vis-à-vis firms active in the same field. Results show that generally firms which had an impact on the development of the technology have very high diversity and there is no trend found which shows any decrease i.e. any specialization of firms. Moreover, similar results are also obtained for the yearly average diversity index of firms which have innovated that year.

Except in few cases it is also found that most cited patents generally date back to 1990s and they are not found on the knowledge flow maps. The reason of this difference comes from the fact that the SPNP algorithm depicts the patents which have a high impact on the development of the technology and also patents which are on the strategic and important junctions of the knowledge flow. Evaluation of patents by their number of citations favors generally older patents. In these samples few patents from video indexing and visual analysis which are found on the knowledge flow maps can also be found in the list of the highly cited patents but these patents are filed before 1990. Moreover, evaluating patents by their number of citations cannot give an overview of the technological development. It is also found that patents on the network of main paths and those which are highly cited patents are not awarded to non-producing entities given by Allison et al. (2009) and by Bessen et al. (2011).

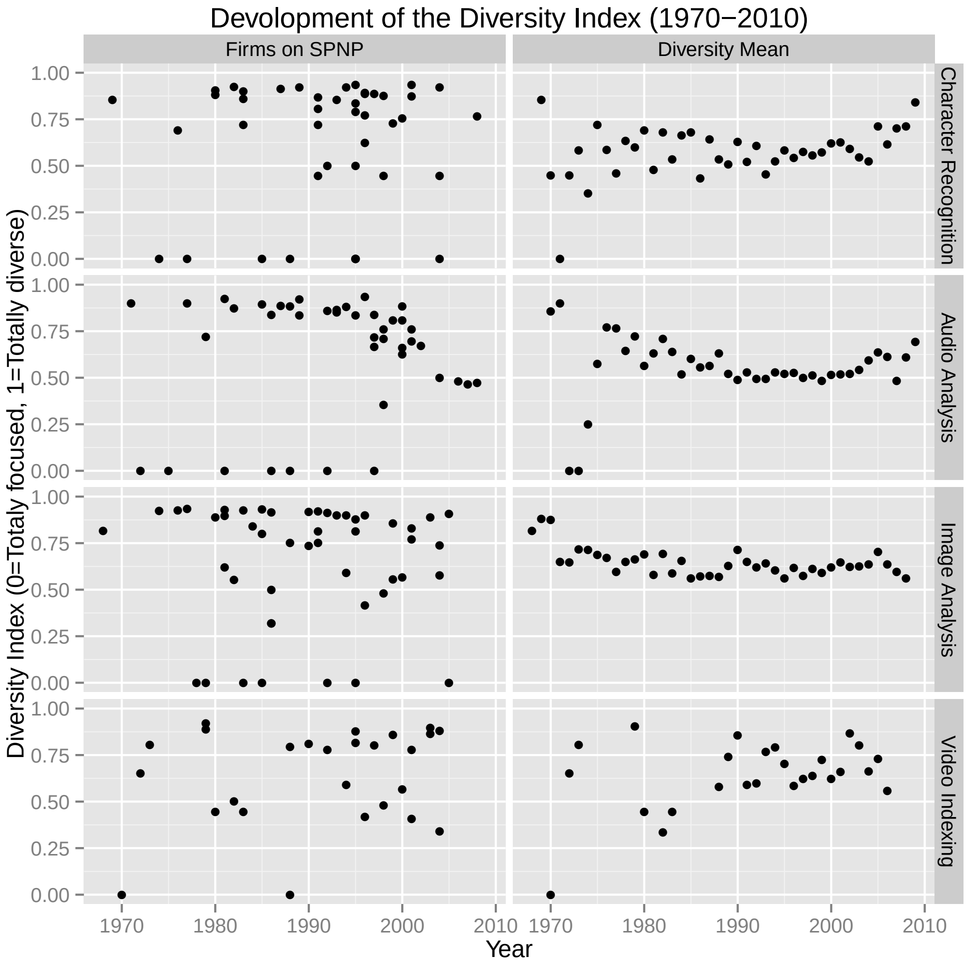


Figure 5.9: Diversity index of firms which have made technological achievement (left column) and the diversity mean of firms (right column).

5.6 Discussion

In this chapter, the knowledge boundaries of firms which have made technological achievements in video indexing are investigated. Video indexing is a modular technology and theory on modularity is used to interpret the results obtained. There are two competing points of view in the innovativeness of firms; one argues that technologically diversified firms are more innovative through cross-breeding of ideas, the other claims that firms specialized in few

R&D area are more innovative (Brusoni et al., 2001).

In software production modularity is an important tool to overcome complexity and the increasing number of features added. The theory on modularity argues that modularity has a significant impact on production but also on organizations. There are two divergent arguments within the theory of modularity. Langlois and Robertson (1992) argue that modularity will result in vertical and horizontal disintegration. On the other hand, Schilling (2000) argues that not all systems evolve into modularity and it is possible that some systems would eventually move towards higher integration. The aim of the previous and this chapter is to find whether firms making significant technological achievements in software technologies and their related industries move towards technological diversification or technological specialization.

Patent connectivity analysis of video indexing and its three modules provide arguments that some multi-technology firms could conduct a vertical integration in video indexing. Even though there are firms having patents on numerous networks of main paths, there is one firm (Siemens AG) which has patents on all of the four networks of main paths. It is possible that this firm has a vertical integration in video indexing and it is also found that Siemens AG is active in large scale surveillance technologies.

Firm is the unit analysis in the left hand side of the Figure 5.9. Each dot represents a firm which has made a technological achievement for the module at the year given in the horizontal axis. Figure 5.9 shows that firms with a very low diversity index focus solely on a distinct technology. It is found that most of these firms have few patents. These firms could provide complementary technologies, modules which could be deployed by the system integrator firms in video indexing. These results cannot help us to conclude that these firms, which are generally not specialized, adopt a vertical integration in any module analyzed. A more in-depth analysis of delivered products could give some insight whether these firms adopt a vertical integration as a mean for rent appropriation (Teece, 1986) or if there is a trend of horizontal and vertical disintegration of firms as Langlois and Robertson (1992) argue.

It is found that firms which have diversity index above the average diversity of their respective module had made significant technological achievements as shown in Figure 5.9. However, video indexing does not follow this pattern. Many of the firms contributing to the development of the video indexing are bellow the diversity index of the video indexing patent group. Moreover, the right hand side of the figure points that the mean of the diversity index of firms is also not very stable in the case of the video indexing.

In rapidly changing technological environments firms focus on their core technologies while depending on others for complementarity (Garud and Kumaraswamy, 1995). The theory of software modularity argues various advantages of such software architecture (Parnas, 1972). It is also shown in few examples that software evolves into an increased modularity (MacCormack et al., 2006; Fortuna et al., 2011). Moreover, it is argued that modularity results in an increasing specialization (Langlois and Robertson, 1992). However, results show that firms which have contributed to the development through technological achievements to the character recognition, audio analysis and image analysis, except a number of few, are highly diverse. In all of the four software modules it is not possible to find any distinct trend toward specialization. This high diversity cannot be explained with the technological development phase because the analyzed data frame is from 1970 to 2010. During this time frame in all three modules diversity index values of firms which have made technological achievements and the diversity mean of the modules related industry do not change much and their levels are more or less constant. In this analysis merging and acquisition, if occurred, are not taken into account and the name of firms provided by the PATSTAT database is used.

Conway (1968) argues that the software modularity should also have a copy of development teams' communication channels. The unit analysis in this study is firm and the inner organizational structure is not analyzed. Thus, even if the video indexing is a highly modular technology it is not possible to trace the organizational modularity within the analyzed firms and their communication channels between teams which are developing the necessary modules.

5.7 Conclusion

The patent analysis presented in this paper is made of two steps to understand the technological development of a modular software technology. First, the product architecture of the video indexing technology is set to determine keywords to be used in patent search. Second, from this product system architecture, three other distinct technologies are identified. These technologies are in a more advanced development phase compared to video indexing. The three technologies; character recognition, audio/speech analysis and image analysis are analyzed through patent analysis. The aim is to find if an increasing modularity involves an increasing specialization of firms.

Five basic results are found. First, different technological branches for each of the three technologies are depicted. Second, patents which are found to have an impact on the technological development, i.e. patents on the network of main paths, are not filed by any of the non-practicing entities. Third, it is found that firms having an impact on the development of the technology are either multi-technology firms or in a limited number of cases, they are very specialized. Fourth, there is not any trend towards specialization among firms which have made technological achievements, besides firms active in the same sector also do not show any similar trend. Finally, firms which have made technological achievements have higher diversity index then the average of the firms of the same sector.

It is also found out that firms contributing to the technology have a steady diversity index level, except the video indexing. The reason is that video indexing is not at a very advanced level compared to its well established modules. According to our results, firms which made technologically significant achievements are characterized by diversity.

Chapter 6

Sharing and excluding: The double life of firms contributing to the Linux kernel project

The innovation model and the regimes of appropriation underlying free and open source software (FLOSS) development differ from that of other technical domains. FLOSS development model is distributed and participatory with source code open to anyone. On the other hand, proprietary software development model is closed from its inception until its delivery. Patents are also an important IPR management tool that are extensively used by corporations to protect their inventions. Though, FLOSS property scheme is "around the right to distribute, not the right to exclude" (Weber, 2004). This chapter aims to contribute to the literature by examining the impact of patenting and FLOSS development activities on performance of firms through econometric evidences.

Since the 1990s FLOSS development and use gained an important momentum. At the same time a patent explosion is also observed. This increase is attributed to various facts such as the research performance of electrical, electronics, computing, and scientific instruments industries (Hall, 2004), the new innovation of management practices (Kortum and Lerner, 1999), the investment of universities and public research institutes into patenting

(Mowery, 2001) and the strategic patenting of firms (Shapiro, 2001; Bessen and Hunt, 2007).

There is an important debate on the necessity of software patents. FLOSS community is opposing to software patents but there has always been a discussion on software patents within the literature. It is argued that software patents deter innovativeness of firms (Nelson, 1994; Bessen, 2009; Bessen et al., 2011). In addition, it is suggested that a tight intellectual property right (IPR) regime would increase the number of patent litigation (Dosi et al., 2006), which is subsequently demonstrated for software patents. It is shown that 94% of the patent lawsuits are related to software patents (Allison et al., 2009). Within this legal environment, many FLOSS advocates and software developers claim that software patents become an important threat to the development of FLOSS. Moreover, the existing literature on innovation offers different arguments in opposition to the software patents (Nelson, 1994; Hunt, 2001; Hunt and Bessen, 2004; Bessen, 2009; Bessen et al., 2011).

Contrary to the FLOSS advocates, there is not a clear cut limit between firms working in FLOSS and proprietary software domain. Most of the FLOSS projects started either as an academic project or as a personal curiosity then some of them changed into a sponsored project or continue to be developed within firms. The Linux kernel project operating system started in 1991 as a student project targeting only Intel x86 based PCs. Now, the Linux kernel project finds many different application areas. While the project size matters (number of contributors, projects popularity, technical difficulties that the project aims to solve) it has been shown that the motivations of the developer communities are heterogeneous (David and Shapiro, 2008). In addition, FLOSS projects are a skill development environment even for those with prior knowledge (Glott and Ghosh, 2005). On the other hand, it is observed that there are more and more contributions from firms to FLOSS projects. Firms contributing to FLOSS projects have patent portfolios containing software as well as non-software patents. Moreover, firms like RedHat, which have started as a FLOSS based company, are also filing various patents. It is found that firms with large stock of software patents are more likely to release FLOSS products (Fosfuri et al., 2008).

It is demonstrated in previous research that firms entering into the FLOSS field adopt a hybrid business model which is a combination of proprietary and FLOSS product development under different licensing schemes (Bonaccorsi et al., 2006; Dahlander and McKelvey, 2005). It is also shown that 75% of the Linux kernel which is one of the successful flagships of the FLOSS movement is developed by those being paid for their work in the project (Corbet et al., 2012). These results show that at least in the case of the Linux kernel, which was at its debut a hobby and a student project, became mainly a corporate product licensed under the General Public License (GPL) which aims to protect any appropriations of the code. GPL ensures that the code remains open for future releases and if the code is distributed, its source code should also be provided (Stallman, 2002). The question arises whether the contribution to FLOSS and patenting are contradictory for these firms.

In relation to these observations on FLOSS and software patents, this chapter will focus on the following questions; how contribution to the Linux kernel project and patenting contributes to the firm performance and are these two activities complementary?

Three databases are used in this research. EPO (European Patenting Office) PATSTAT (2012) April edition is used to obtain patents of firms contributing to the Linux kernel project. Firm level data such as R&D expenditure, number of employees, sales, etc., are obtained from Thomson One Banker. And finally, data related to the contribution to the Linux kernel project is obtained from The Linux Foundation. Among over 800 firms which have contributed to the project for the seven years period (2005-2011), firm level data for 169 firms are analyzed through panel data econometrics. Contribution to the Linux kernel project is measured by the number of the *changeset* (the group of modification containing files which are relevant to each other). The analyzed 169 firms contributed to the Linux kernel with 48.7 % of the total changeset during these seven years. The results obtained in this chapter show that the general patenting and contribution to the Linux kernel project affect positively sales of the firms. However, their interaction has a negative effect on the performance of the firms.

The next section presents the background on FLOSS, the Linux kernel development and software patents and Section 6.2 proposes hypotheses tested in this chapter. Section 6.3 gives an overview of the data. Section 6.4 presents and discusses the results of the analysis, and the last section concludes.

6.1 Background on IPR and FLOSS

Intellectual property rights (IPR) regimes defended within FLOSS community are on the opposite sides of the proprietary software development model and of the patent system. FLOSS development has traditionally been associated to the "hacker" ethic which is defined as early as 1980s. Building blocks of the hacker ethics are; access to computers, free information, mistrust to the authority and promotion of decentralization, evaluations based only on hacking capabilities, search of art and beauty on computers and better life through computing (Levy, 2010, ch.2). Within this stream, licensing schemes, which are developed for FLOSS, are enforcing sharing and encouraging the study of the software source code. On the other hand, a patent gives a monopoly right to the inventor by excluding others to use a particular invention for a limited time. The exclusion argument is completed by Cohendet and Pénin (2011) with the assertion that patents facilitate coordination between firms by encouraging the emergence of new markets and acting like a non-market coordination force through inter-firm collaborations.

Most often, patents are explained with their aim which is to encourage innovation (Hall, 2007). It is also argued that the resulting effects of the patent system are not the same for all technologies. In some sectors such as chemical, pharmaceutical and biotechnology industry, it is argued that patents could promote innovation (Arora et al., 2001). However, it is also argued that strong patent protection has a negative effect on innovative activities in "cumulative system technologies" such as software, radio and aircraft technologies (Nelson, 1994; Mazzoleni and Nelson, 1998b). The particularities of these technologies are that they are built on top of previous inventions and a single product contains a large quantity of patents. Thus, a product based on cumulative system technologies contains numerous patented inventions. Given the strong IPR regime, patents are one of the barriers to innovate in cumulative system technologies especially for independent software developers due to the risks of patent infringement cases. It is also suggested that strong IPR regime did not play an important role in the emergence of ICTs and it is not a tool for value generation. Dosi et al. (2006) claim that weak IPR regimes played an important role in the emergence and diffusion of transistor, semiconductor and mobile telephony.

The USPTO accepts software patents since 1981 but it has been argued that software industry has historically a weak patent protection regime (Bessen, 2009). On the other hand, even though EPO rejects patents related to computer programs as they are, the European patent legislation does not restrict software related patents which are increasing with time (Rentocchini, 2011).

According to one research stream, patenting increases costs of imitation (Mansfield et al., 1981). But in the case of software patent, it does not only increase costs of imitation but it also creates barriers to the following research in a strong IPR regime (Dosi et al., 2006; Bessen and Hunt, 2007). Moreover, strategies like patent thicketing (Shapiro, 2001) within a cumulative system technology in a strong IPR regime creates considerable amount of legal complexities in industries such as software. Patent lawyers and non-practicing entities (NPEs or patent trolls) become important figures in the software industry due to the increasing number of patent litigation cases. It is ironically pointed out that the number of patent lawyers is growing faster than the amount of research (Barton, 2000). Moreover, as indicated before, it is reported that 94% of the patent lawsuits are related to software patents (Allison et al., 2009). NPEs, firms which have based their business model on licensing and litigating over patents with unpredictable boundaries, have caused an important financial burden to the defendant firms. NPEs are mainly active in software related patents. It is calculated that there is approximately half a trillion dollar loss of wealth from 1990 through 2010 for the defendant technology firms (Bessen et al., 2011). According to the authors, this loss harms society and there is little evidence of money transfer from non-producing entities to independent inventors after litigation.

In the next section different licensing schemes in FLOSS is presented.

Then the relation between FLOSS and patenting is given, followed by a short presentation of the Linux kernel project.

6.1.1 Licensing scheme in FLOSS

The licensing schemes of the FLOSS ensure that the source code is kept in the public domain. Among various FLOSS licensing scheme, GPL is one of the most popular one (de Laat, 2005). GPL is in its third version and keeps evolving with the new development of the software industry. The GPL gives roughly four freedoms to the owner of the code; 1) the freedom to run the software for any purpose, 2) the freedom to read its source code, 3) the freedom to change it and 4) the freedom to give the code to anyone. Yet, any redistribution of the software whether modified or not should be done with the source code (Stallman, 2002). GPL license family (GPLv2, GPLv3, LGPLv2, and LGPv3) usage is about 54% of all software packages released under FLOSS license (Aslett, 2012). GPL is developed by the Free Software Foundation (FSF) which also provides various software to the FLOSS world since 1984.

Another popular FLOSS compliant licensing scheme is the Berkeley Software Distribution (BSD) license which allows an open source code to be distributed in a closed form without providing the source code. This licensing scheme allows many BSD licensed software to be modified and distributed like any proprietary software without releasing the modified source code. The BSD license is considered as a permissive license. The case in this chapter is the Linux kernel project which is only licensed with GPLv2 (Laurent, 2008).

GPL is a strategic move of FLOSS community which circumvents the IPR system to protect users. This is contrary to all other proprietary software licenses which aim to protect the developing firms. This strategic move is further stretched to the patents with the last version of the GPL. GPLv3 is released in 2007 and it is not compatible with patents. It contains several clauses which do not allow the development of free software requiring any patent license. The aim of these clauses is to avoid any legal battle that a FLOSS user or a developer can encounter. This limits any FLOSS contributor firm to release any code which contains patented inventions and use it later against an entity using the the same software. According to some lawyers, GPLv2 implicitly state that a company providing software licensed under GPL cannot sue any user on the same grounds. On the other hand, the BSD license does not have any specific clause on patents.

The Linux kernel adopted GPL since its debut and is developed with the GPLv2. The change of the licensing of a FLOSS project requires the approval of every contributor and also the endorsement of this change by the project leader. If there are contributors who are missing or refusing the change, then their code should be removed and a new one should be rewritten in order to make the transition to a new license. In the case of the Linux kernel project, which has a very high number of contributors and a very long development history, it is very difficult to contact every contributor and obtain their approval. It is difficult to track the author of every line of code and ask their permission as the code is also changing in every release. Moreover, this process should be very problematic for older contributions. Furthermore, the project leader of the Linux kernel project does not approve several clauses within GPLv3.

The opposite of the FLOSS software is the propriety software, which in general, does not provide any access to the source code. Moreover, it is forbidden to redistribute, modify and reverse engineer a proprietary code. The source code is protected under exclusive legal right of the copyright owner.

6.1.2 FLOSS and patenting

It is reported that software patents are used strategically by established firms to build thickets to restrain competition to obtain a competitive advantage (Bessen and Hunt, 2007). Another strategy is to add market value to the company through patenting for a possible acquisition. When the value of software patents is questioned, it is not possible to find a clear answer if those patents increase the value of a pure software firm within markets (Hall and MacGarvie, 2010). In spite of that, patenting have a positive effect on the IPO values of software firms in European and US market (Useche, 2012).

There are three groups within the FLOSS community who argue that software patents are inherently an important threat to FLOSS. The first one is the developers' point of view; it is very unlikely that FLOSS project members check whether the project constituents are infringing any patents. Most of the time a FLOSS project starts because few developers feel the need to develop it due to a "scratching the personal itch" (Raymond, 1999). Independent developers, in case of an infringement suit, are at the mercy of the patent defendant. Most of the FLOSS developers do not have any sufficient resources to confront a legal dispute on a patent litigation case. The second group is the corporate FLOSS users and developers who could face patent litigation cases. Within this second group there are many Fortune 500 firms which represent the wealthiest target for patent trolls. And the last group is composed the end users who also feel the patent threat which could damage the development of their favorite software.

Another strategic move came from IBM, Novell, Philips, RedHat and Sony which created a patent portfolio to protect the development of the GNU/Linux against patent litigation. This consortium is named Open Innovation Network[1] and holds over 600 US patents in which there are several fundamental patents. This consortium is open to other firms and members are bind not to use patent law suits against GNU/Linux users and developers. This consortium can also use their own fundamental patents to counter sue if one member faces a patent litigation case based on their activities with their use of the GNU/Linux project. There is also a distinct group, not tied directly with the Linux kernel project, which defends FLOSS and also the abolishing of the whole patent system.

6.1.3 The Linux kernel project

In this research, revenues and patenting behaviors of firms contributing to the Linux kernel are analyzed. The Linux kernel is the operating system kernel which powers over 90% of the fastest 500 computers in the world in

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

2012 but it is also used in PCs, laptops, smart-phones, routers and many other electronic appliances. It is the main link between the hardware and the applications software. A set of FLOSS software which includes the Linux operating system is called a GNU/Linux distribution. It is calculated that the number of lines of the Linux kernel code represent only 6% percent of a whole set of Debian GNU/Linux distribution released in 2002 (Amor et al., 2005). The Linux kernel project is considered as one of the most successful, a show case among FLOSS projects.

The Linux kernel project receives contributions from different hardware manufacturers and it is installed to the core of their hardware. On the other hand, some other hardware manufacturers refrain to contribute to the project in order not to reveal the internal mechanism of their hardware. They could also prefer not to invest in an in-house FLOSS developer group which would work solely for the hardware produced. In that case, the community could develop FLOSS code in order to run the GNU/Linux on top of the hardware which misses GNU/Linux support. This activity requires reverse engineering and sometimes the result is not good enough if the hardware company could have given at least the blue prints of their hardware. There are many cases in which firms reveal their code selectively. The main reasons of this openness are the enforcement of the GPL, ensuring a positive image of the company within the community and the reduction of the cost of maintenance of the revealed code (Henkel, 2006).

As the project matured and the code base of the Linux kernel project grew, the leader of the project Linus Torvalds could no longer reviewed all patches or contributions made to the project. In order to overcome this work load the project organization has changed in 1998. Besides the leader, other co-maintainers who are responsible for different modules are also involved in the code review and code inclusion to the Linux kernel source code. These co-maintainers create a layer between the project leader and the developer community. They are generally an important developer who has gained the trust of the project leader Linus Torvalds. The leader owns the project, could take binding decisions for the project, work to resolve conflicts among maintainers, but shares the reputation with co-maintainers. Any patch or new code related to a module or a task of the Linux kernel is sent to its co-maintainer who approves or rejects the submitted code contribution (Raymond, 1999).

In this chapter the contributions to the Linux kernel project is measured by the number of modifications on the Linux kernel containing files which are relevant to each other; the *changeset*. If a code or a changeset into the Linux kernel project is accepted it is protected by the GPL, thus, its source code is free and open to everyone. The list of top 15 firms which have contributed to the Linux kernel project between 2005 and 2011 is in Table 6.1.

Table 6.1: List of top 15 contributing firms to the Linux kernel project (changeset) between 2005-2011.

Firm name	# of changeset
Red Hat Inc	30760
International Business Machines Corp.	15943
Intel Corp.	15597
Oracle Corp.	5371
Renesas Electronics Corp.	3298
Fujitsu Limited	3102
Nokia Corporation	3073
Analog Devices Inc	2578
Texas Instruments Inc	2495
Google Inc	2432
Broadcom Corp.	2427
Advanced Micro Devices Inc	2397
Silicon Graphics International Corp.	2395
Cisco Systems Inc	2276
Hewlett-Packard Company	2178

6.2 Previous research and hypotheses

Patents are generally regarded as a reasonable output indicator of successful R&D activities. Other indicators which are used in the literature to measure innovation performance are; R&D input, patent citations and new product announcement (Hagedoorn and Cloodt, 2003). Authors consider that R&D inputs could be used to represent innovative effort, patent citations reflect the quality of patents and innovation activities and product announcements show the product innovation. On the other hand, there is a diverging point of view on the interpretation of patents. In this chapter patents are considered as the output of innovative performance.

Technical achievements which are carried out through R&D are inventions. A firm may patent some of their inventions and commercialize some others. Innovative activities are inventions which are commercialized with the expectation of generating some revenues to the firm. For an innovation to be successful, various resources of firms are also required. Depending on the IPR strategies of firms some inventions, which are patented, do not come to life within a product. In this section FLOSS related IPR business models and associated hypotheses are presented.

An important stream of research on patents investigated the effect of patents on the performances of firms (Ernst, 2001). The criteria of performance of firms are generally sales, profit, profit ratio i.e. percentage of profits over sales, market value of firms. A list of past empirical studies are given by Ernst (2001). It is assumed that patents are a basic proxy for successful R&D and hence the following hypothesis.

H1: Patents applications affect positively performance of firms.

Figure 6.1 is a modified version of the figure given by Basberg (1987) and Ernst (2001). For the sake of clarity on the research questions, activities related to the Linux kernel project is added to the original figure. It shows a conceptual relationship between patents, innovations, inventions and contribution to the Linux project which are considered as the innovation outputs of firms analyzed in this chapter. A firm's contribution to the Linux kernel project is taken as an innovative activity and it is accessible to anyone. Changesets are the contributions of developers, which are approved by the project leader or co-maintainers. Due to the Linux kernel project licensing scheme, these contributions are protected with the General Public License, thus, it is free for anyone to use them, modify them and distribute them. Depending on the industry, patenting or giving away freely some of the innovative activities is the result of a varying patenting and innovation strategies of firms.

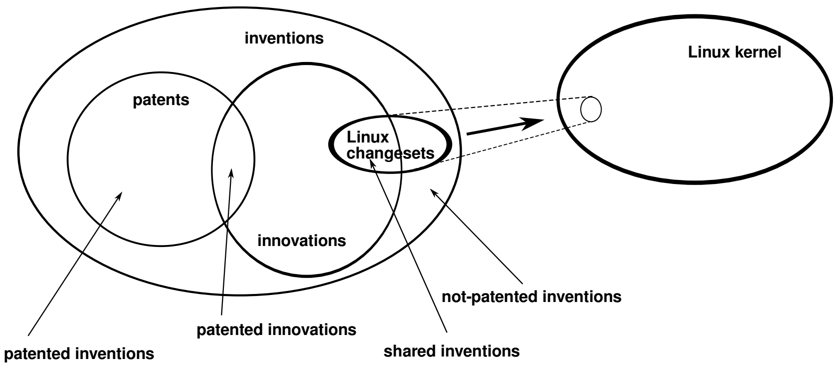


Figure 6.1: Relationship between patents, inventions, innovations and contribution to the Linux kernel project.

Since the early examples of FLOSS literature, it is shown that FLOSS development model is economically viable not only for individuals (Hertel et al., 2003) but also for firms (Hecker, 1999; Lerner and Tirole, 2002). The variations in the involvement of firms in FLOSS is related to the skills of the users on the different markets (Jullien and Zimmermann, 2009). Consequently, since the start of the FLOSS movement many business models are developed to profit from FLOSS. These studies suggest that contribution to the Linux kernel project increases firms revenues.

> **H2:** Contribution to the Linux kernel project affects positively performance of firms.

The hybrid business model consists of revenues generated by the combination of the traditional proprietary software licensing fees and FLOSS related services (Bonaccorsi et al., 2006; Harison and Koski, 2010). The hybrid business model is used as an entry strategy. It is shown that the hybrid business model is not transitional; after its entry, a firm which adopts a hybrid business model does not change it into pure FLOSS development or into a pure proprietary software development model (Bonaccorsi et al., 2006). Among Italian software firms, it is observed that there are lock-in effects for switching to FLOSS on both the supply and the demand side. Even though FLOSS is open to anyone, firms producing FLOSS products and services are driven mainly by economic and technological advantages rather than any altruistic factors. The coexistence of proprietary and FLOSS with different remuneration modes among software developers are interpreted as the key to the success of the FLOSS movement (Bonaccorsi and Rossi, 2003).

Dual licensing is also used by some software developers. This business model is the release of the same software with FLOSS compliant license and with a proprietary software license. Software with proprietary license generally includes some more software which are not released with the FLOSS edition and it can also include some other advantages such as more professional support. Within a game theory approach it is found that dual licensing can be a relevant strategy for software firms (Darmon and Torre, 2010). However, authors argue that it is not always the best approach for the user welfare.

The concept of IP modularity affirms that while some software producers are revealing some of their codes through FLOSS related licensing, they also keep the crucial parts of their software as proprietary. This theory is developed through the examples found within the video game industry. The IP modularity helps firms to better capture value in situations where knowledge and value creation are distributed among many actors. It is argued that IP modularity helps to reconcile co-creation and value capture (Henkel and Baldwin, 2009).

A blend of IP modularity and hybrid business model is observed among the Linux kernel project contributor firms. Some of these firms do not seek a real co-creation but aims to widen their hardware or software user base by providing FLOSS code into the Linux kernel while keeping some of their code and knowledge proprietary through different means. Firms knowledge sharing is selective as it is shown for the case of the embedded Linux (Henkel, 2006). In many cases firms contribute to the development of the Linux kernel but they also provide some other complementing parts with proprietary software which has restrictive license terms and without source code.

One of the typical examples is seen within the Nvidia and ATI dominated graphics processing units (GPU) market. In some cases video card manufacturers give away FLOSS licensed drivers which lacks proper 3-D support.

For an optimum result again a proprietary driver might be needed. On the other hand, these firms have a substantial number of patents related to their hardware. Some of the reasons of selective code revealing, as in the case of the embedded Linux, are expectations of having a positive image of the company within the community and the reduction of the cost of code maintenance (Henkel, 2006). While they reveal and share some of their knowledge through their contribution to the Linux kernel project, firms also patent some of their inventions to protect their intellectual properties. Both activities go hand in hand according to the innovation strategies of firms. The interaction effect is measured to understand whether the marginal returns of one variable increases the level of the second variable when they occur together. The result of the interaction will provide whether there is a complementarity or substitution.

> **H3:** There is a positive interaction effect between the contribution to the Linux kernel project and patenting as far as their effect on the performance of firms is concerned.

6.3 Data and econometric model

The data on the contribution to the Linux kernel project consists of number of changeset given by firms during seven years, from the start of 2005 to the end of the year 2011 ($T = 7$). During these seven years there are 32 Linux kernel releases, one release for every 8 to 12 weeks are observed. During this time period over 800 different firms have contributed to the project. Figure 6.2 shows contributing entities with aggregated contributed changeset values for "Firms", "None", "Unknown", "Consultants" and "Public institutes". "None" represents developers who are known to contribute without any salary from any firm, they are volunteers. "Unknown" represents case in which the affiliation of the contributor is not found. "Public institutes" comprise universities, research institutes and contribution made from developers working in governmental departments. This grouping is made by using the extension of the email address of the developers and if the developer provided their affiliations[2].

Among over 800 firms, entry of 360 firms are found in the Thomson One Banker but only 169 of them comprise data on characteristics of firms ($N = 169$). These 169 firms are denoted "Analyzed firms" and their contribution to the Linux kernel project is shown on Figure 6.2. This group of "Analyzed firms" is a subset of the group "Firms" on that figure. Analyzed firms are contributing roughly to the half of the Linux project. Thomson One Banker provides sales revenues in different currencies which are changed to USD through OECD data and for three non-OECD countries `oanda.com` is used. To check the correctness of the processed firm level data `ychart.com` is also used but in case of differences, data provided by Thomson One Banker is taken into account.

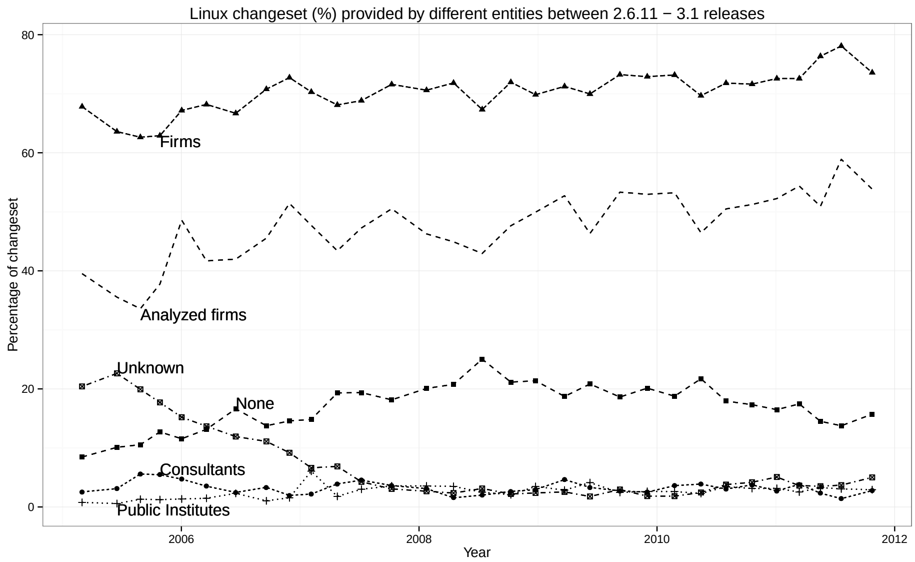
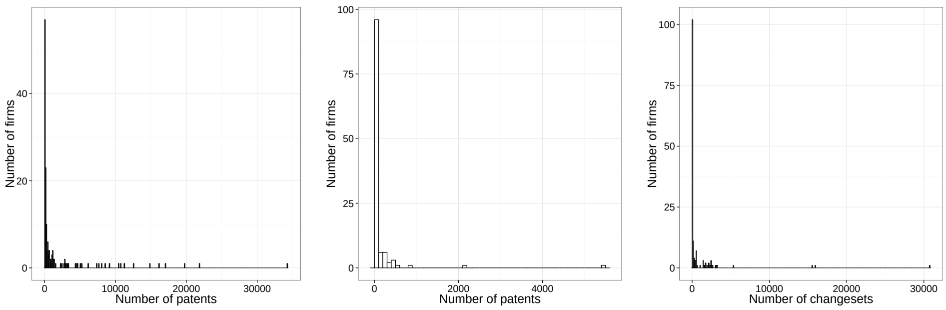


Figure 6.2: Percent of changeset contribution of the five entities.

The number of firms analyzed in this study is 169 for the period of 2005-2011. The number of patent application is used to reflect the innovation

[2]Raw data on contributors are retrieved from `https://github.com/gregkh/kernel-history` on August 2012 from kernel developer Greg Kroah-Hartman's repository.



(a) All patents (b) Software patents (c) Changesets

Figure 6.3: Distribution of patent applications and changesets among the analyzed firms. All histograms use a discrete interval = 100.

output of firms. For the period of 2005-2011, among these 169 firms 146 of them published patents and 109 of them contain at least one software patent. The distributions of patent applications is given in Figure 6.3(a), Figure 6.3(b) shows the distribution of software patents and the contributed changeset is given in Figure 6.3(c). The number of patent applications and changesets are summed for each year for the analyzed firms along seven years.

For the regression analysis, patent statistics are obtained from EPO PATSTAT 2012 April edition, and only patent applications made to the USPTO are evaluated. Patent application numbers represent the sum of non-software patent and software patent applications. Software patents do not have distinct IPC codes which make it difficult to evaluate statistically their number. In this chapter the number of software patent applications made by each firm is obtained through the method used by Hall and MacGarvie (2010). In this method software patents are obtained by the intersection of two sets of patents which are constructed through the definitions of software patents given in several previous researches. The first set is obtained by the definitions of Mowery and Graham (as cited in Hall and MacGarvie (2010)) and Hall and MacGarvie (2010). The second set is obtained by the search criteria given by Bessen (2010). The first set is the sum of the patents in "Electric Digital Data Processing" (G06F), "Recognition of Data; Presentation of Data; Record Carriers; Handling Record Carriers" (G06K) and

"Electric Communication Technique" (H04L) IPC classes (Mowery and Graham, 2003) and all patents in IPC classes of patents awarded to the fifteen of the largest US software firms given by Hall and MacGarvie (2010). The second group is made of patents obtained by the method used by Bessen (2010) which is adapted to the PATSTAT database. This method consists of keyword search on title and abstract. Keywords used are "software" or the words "computer" and "program" in abstract but these patents should not contain "semiconductor", "chip", "circuit", "circuitry" or "bus" in their title. Patents containing "antigen", "antigenic", or "chromatography" in patent abstracts are also excluded. From this set of software patent applications, applications of firms which are active in the development of the Linux kernel are counted.

From these data various tables are generated for the period of 2005-2011. Table 6.2 gives the list of the top 15 USPTO patent recipient among firms contributing to the Linux kernel project. In Table 6.3 the list of the top 15 USPTO software patent recipient firms found within the Linux kernel project contributor firms are shown.

Table 6.2: List of top 15 USPTO patent recipient firms between 2005-2011 among the Linux kernel contributor firms.

Firm name	# of patents
International Business Machines Corp.	30813
Hitachi Limited	17258
Microsoft Corp.	14153
Sony Corporation	14042
LG Corp.	11577
Intel Corp.	11563
Fujitsu Limited	11351
Hewlett-Packard Company	10853
Siemens AG	9361
General Electric Company	7467
NEC Corporation	7304
Panasonic Corporation	7098
Xerox Corp.	7001
LG Electronics Inc	6607
Texas Instruments Inc	5491

The number of patents and contribution to the Linux kernel project vary across industries and time due to differences in the factors affecting the de-

Table 6.3: List of top 15 USPTO software patent recipient firms between 2005-2011 among the Linux kernel contributor firms.

Firm name	# of patents
International Business Machines Corp.	3335
Microsoft Corp.	1371
Hewlett-Packard Company	402
Nokia Corporation	317
Intel Corp.	285
Sony Corporation	244
Hitachi Limited	235
Oracle Corp.	229
Siemens AG	228
Cisco Systems Inc	190
Fujitsu Limited	180
Symantec Corp.	152
The Boeing Company	133
Broadcom Corp.	127
Apple Inc	117

cision to patent and contribute to the Linux kernel project. Table 6.5 gives the distribution of firms analyzed in this chapter according to their Global Industry Classification Standard (GICS) code. The group named "other" is the aggregation of industries such as transport, media, capital goods, automotive, health etc. There are also 18 firms which do not have any GICS code.

6.3.1 Model

Sale is the dependent variable and the effect of inventions of firms on the sales of firms is evaluated. From the Figure 6.1, it is argued that the number of patents and number of Linux changeset provided by firms are values which reflect their intensity of invention. It is shown from various previous researches that the number of patent applications has positive effect on sales of firms (Ernst, 2001).

The sample is very skewed and heterogeneous in patenting and contribution to the Linux kernel project. This fact forces to use a simple first order log-log model. Thus the basic model used in this chapter is as follows:

$$log(sales)_{it} = \alpha_t + \gamma log X_{it} + \sigma log Z_{it} + T_{it} + \xi_{it} \tag{6.1}$$

Table 6.4: Variable definitions.

	Variable	Definition
dependent	*sales*	sales in USD.
independent	*yearly_(non_)soft_pat*	number of (non) software patent application made each year.
	changeset	group of modification on the Linux kernel containing files which are relevant to each other.
control variable	*rd_expense*	R&D expenses.
	changeset_stock	stock number of changeset since 2002, with 15% depreciation.
	(soft_)pat_stock	(software) patent stock calculated starting 1990, with 15% depreciation rate.
	dummy_changeset	1 if there is at least one changeset given by the firm i at year t else 0.
	dummy_(soft_)patent	1 if there is a (software) patent application made by the firm i at year t else 0.

Table 6.5: Industry distribution according to the GICS codes of the analyzed 169 firms.

Sector	# of firms
Hardware	56
Semiconductor	44
Software	31
Other	38
Total	169

where X_{it} is a vector of the main exploratory variables: the number of patent application, the number of changeset and their interactions. Z_{it} is the vector of the control variables such as stock of patent applications, the stock of the Linux contribution etc. T_{it} is a dummy for years, ξ_{it} captures the specific residuals. Fixed effects model is used in order to control for unobserved heterogeneity across firms over time. According to the sample used in this chapter, fixed effects models are found to be superior to random effects models based on the Hausman (1978) test. The Breusch and Pagan (1979) test for heteroskedasticity is carried out and heteroskedasticity consistent coefficients are given in Table 6.8. All values which are (0) are changed to (0.5) so that their log values are not minus infinite (Cameron and K.Trivedi, 2005).

R&D expenditure indicates the innovative competence of firms particularly in high-tech industries (Hagedoorn and Cloodt, 2003). The loss in the

Table 6.6: Descriptive statistics.

	N	mean	sd	min	max
log(sales)	1120	20.93	2.49	12.35	26.05
log(yearly_non_soft_pat)	1122	2.85	2.59	-0.69	8.97
log(changeset)	1122	1.27	2.47	-0.69	8.82
log(rd_expense)	1048	18.29	3.77	-0.69	22.93
log(yearly_soft_pat)	1122	0.41	1.63	-0.69	7.34
log(pat_stock)	1122	3.60	3.09	-1.47	10.10
log(changeset_stock)	1122	1.67	2.81	-0.69	9.97
dummy_patent	1122	0.82	0.39	0.00	1.00
dummy_soft_patent	1122	0.44	0.50	0.00	1.00
dummy_changeset	1122	0.52	0.50	0.00	1.00

Table 6.7: Correalation matrix.

		1	2	3	4	5	6	7	8	9	10
log(sales)	1	1									
log(yearly_non_soft_pat)	2	0.717	1								
log(changeset)	3	0.266	0.333	1							
log(rd_expense)	4	0.635	0.547	0.223	1						
log(yearly_soft_pat)	5	0.596	0.769	0.372	0.429	1					
log(pat_stock)	6	0.672	0.803	0.343	0.533	0.633	1				
log(changeset_stock)	7	0.265	0.318	0.962	0.201	0.341	0.355	1			
dummy_patent	8	0.4	0.647	0.186	0.387	0.321	0.534	0.186	1		
dummy_soft_patent	9	0.508	0.686	0.289	0.376	0.766	0.564	0.265	0.419	1	
dummy_changeset	10	0.207	0.218	0.756	0.151	0.215	0.254	0.768	0.149	0.173	1

stock of knowledge due to time is also considered (Hall et al., 1986). The knowledge stock is calculated using the usual 15% of depreciation rate over years. In equation 6.2 an example of the knowledge stock with depreciation depending on the R&D spending is shown.

$$K_{it}^{R} = (1-\rho)K_{i,t-1}^{R} + R_{it} \tag{6.2}$$

where R_{it} is the R&D spending for the year t for the firm i and the ρ is the 15% depreciation rate. K_{it}^{R} is the stock of knowledge which increases every year with the investments but there is also a loss in the previous accumulated knowledge. Similar calculations are done for the knowledge stock based on patent numbers and on the Linux contributions. The number of changeset accepted by the project leadership is used to measure the Linux knowledge stock. Finally, to calculate the R&D stock, the yearly R&D expenses in USD values are used. The R&D stock is calculated starting from year 2002, the patent stock is obtained from data starting from 1990 and the knowledge stock on Linux starts from 2005. If the firm is established later than these dates necessary calculations are done based on their establishment year. There is no lagged parameter in the model. It is assumed that the knowledge stock which is measured with patent numbers and the given changesets have an immediate effect on the sales value of firms in such a fast paced industry. Moreover, the knowledge stock comprises past acquired knowledge.

Table 6.8: Estimates for the dependent variable log(sales), fixed effect.

	Model 1	Model 2	Model 3	Model 4
log(yearly_non_soft_pat)	0.11***	0.12***	0.11***	0.12***
	(0.03)	(0.03)	(0.03)	(0.03)
log(changeset)	0.02	0.05*	0.03	0.03
	(0.01)	(0.02)	(0.02)	(0.02)
log(rd_expense)	0.10*	0.10*	0.10	0.10*
	(0.05)	(0.05)	(0.05)	(0.05)
log(yearly_soft_pat)	0.00	0.00	0.00	−0.02
	(0.03)	(0.02)	(0.02)	(0.03)
log(changeset):log(yearly_non_soft_pat)		−0.01*	−0.01*	−0.01*
		(0.00)	(0.00)	(0.00)
log(pat_stock)			0.07*	0.07*
			(0.03)	(0.03)
log(changeset_stock)			0.02	0.02
			(0.03)	(0.03)
factor(dummy_patent)1				−0.10
				(0.07)
factor(dummy_soft_patent)1				0.07
				(0.05)
factor(dummy_changeset)1				0.04
				(0.05)
year dummy	*yes*	*yes*	*yes*	*yes*
R^2	0.29	0.30	0.31	0.32
Adj. R^2	0.24	0.25	0.26	0.26
Num. obs.	1047	1047	1047	1047

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

6.4 Results and discussion

In this chapter, it is assumed that patenting is a proxy for IP management and product development model which are opposite to the Linux kernel project contribution. Patenting is also divided into two different categories as software patents and non-software patents. Data for 169 firms are used in this analysis. These firms have contributed 48.7% of the changesets accepted by the Linux kernel project during the seven years period. Results given in Table 6.8 show that there is not any significant correlation regarding software patents to sales. However, the number of non-software patents and patent stock and R&D expenses are correlated with sales. Moreover, contributing to the Linux kernel project also has positive return to the performance of firms in some models. These results confirm the first and the second hypothesis.

The third hypothesis is framed based on the theory of hybrid business (Bonaccorsi et al., 2006; Dahlander and McKelvey, 2005) and IP modularity (Henkel and Baldwin, 2009). However, there are some differences between software firms which are using hybrid business models or IP modularity with firms analyzed in this chapter. These differences are based on technology and IP management. First, firms which are analyzed in this chapter are only those contributing to the Linux kernel. Contributing to the Linux kernel project requires an important technical knowledge compared to most of the other FLOSS projects. The Linux kernel is a layer between hardware and application software. The complexity of an operating system development comes from the requirement of having knowledge on hardware and on software. Hardware and semiconductor firms are inevitably in need of the whole Linux kernel project to run their hardware. Moreover, many peripheral manufacturers are also contributing to the project if they need their hardware to run on top of the Linux kernel. It is also important for some specialized software such as database to run on a fine tuned operating system. Second, firms which have adopted hybrid business model are giving services and software products in proprietary software model and also contributing to FLOSS. Firms analyzed in this chapter are patenting and contributing to the Linux kernel project.

The interaction of patenting and contribution to the Linux kernel project on sales are found to be negative. These two activities are not complementary but they are substitute. According to the sample, it is concluded that as the number of non-software patents increases, sales also increase. Thus, it is found that the marginal effect of patents is not diminishing but their interaction with the contribution to the Linux kernel project lessens the sales. This result contradicts the third hypothesis that the interaction of contributing to the Linux project and patenting would increase sales of firms. This negative interaction is interpreted in the light of intellectual property rights and firm level innovation strategies.

Advantages of using GNU/Linux system as presented by Varian and Shapiro (2003) are accepted more within the business circles. Consequently, its usage also increases. Various firms which are active in different market also endorse the development of FLOSS. Figure 6.2 shows that the Linux kernel project development is gaining traction during which the number of firms contributing to the Linux kernel project increases.

Henkel (2006) conducted research gives some leading path for the interpretation of the negative interaction. The author conducted a survey among embedded Linux developing firms. Embedded Linux is a subset of the Linux kernel project analyzed in this chapter. Henkel (2006) shows that embedded Linux producers revealed code is generic and it can be used by others. Moreover, these firms are contributing to the project because of the requirements of the GPL but also to have a positive reputation within the community. The same research reveals that another advantage of contribution to the Linux kernel project is to set the development trajectory while leaving the development of the code to the community.

Firms contributing to the project and then leaving it to the community to continue on its development like in the case of the embedded Linux Henkel (2006) explain the negative interaction. Firms support the development of the project even it is to their disadvantage when coupled with their patenting activities but in the long run the support obtained by the community explain their initial support. Moreover, a second interpretation of the result is tied to the fact that firms which use a hybrid business model are locked as in an incompatible innovation model. This reminds the Italian software firms which cannot change to pure FLOSS or to pure proprietary software model for product and/or service delivery after having adopted hybrid business model Bonaccorsi et al. (2006).

Previous researches show that software patents come into play when firms need to secure some funding for investment (Hall and MacGarvie, 2010; Useche, 2012). In this research, there is not any significant result showing a correlation between software patents and sales. It could also be expected that firms contributing to the Linux kernel project file software patents for defensive and strategic purposes. Software patents are an intriguing subject within the FLOSS eco-system where FLOSS firms are also forming patent alliances for defensive purposes. On the other hand, firms such as Microsoft is profiting the most from the Android operating system, a derivative work of the Linux kernel project lead by Google for smartphones and tablets. Microsoft is profiting from the Android based phone sales through licensing some of its key phone related patents. As Android phone sales increases Microsoft generates revenues through licensing[3].

As a robustness check, same models found in Table 6.8 are used by removing top three non-software and software patent recipient firms, as well as the top three changeset providers to the Linux project. The lists of these firms are found in Tables 6.2, 6.3, 6.1. By removing these firms, it is aimed to lessen the skewness of the analyzed data. Firms which are removed from the panel data set are; IBM, Hitachi, Microsoft, HP, RedHat and Intel. Results obtained are similar to those shown in Table 6.8.

6.5 Conclusion

This chapter studied the effect of patenting and contribution to the Linux kernel project and their interaction on sales. It is found that patenting and contribution to the Linux kernel project affects sales positively. However, their interaction contributes negatively to the firm performance showing

[3]http://www.zdnet.com/microsoft-profits-from-linux-patent-fud-7000001598/

a substitution between these two innovation outputs. These firms develop products by excluding others through patenting and at same time contribute to the development of the Linux kernel which has a source code open to anyone. Results are analyzed through IPR and and firm level innovation strategies.

Results reveal that firms are trenched in their IPR management. Firms have developed a strong routine to protect their innovations through patenting, but at the same time advantages to give away some of their innovation outputs are also taking root. It is argued that the contradiction of patenting and contributing to the Linux kernel project show that there is a double faceted innovation model adopted by firms analyzed in this chapter. It is concluded that analyzed firms are locked-in in two different innovation model but they also rely on the support given by the community and other returns such as the positive reputation obtained from community; code developers as well as users.

One of the problems with such studies is the use of aggregate data. In the next chapter, to complete this chapter and to overcome the critics of the aggregate data usage, a case study is conducted. In the next chapter, difficulties of adoption of open innovation model by a software R&D group are studied.

Chapter 7

Adopting open innovation by a software R&D group: A case study

In the previous chapter, it is found that unveiling and sharing some results of innovation by contributing to the GNU/Linux kernel project and patenting are found to be correlated positively with sales. However, their interaction is found to be negative showing a substitution. It is concluded that these results are linked to strategic IPR management and selective code revealing.

The previous work was carried out with aggregate data of 169 firms contributing to the GNU/Linux kernel project. However, each firm has some peculiarities and case studies could provide detailed insight on software development and IPR issues through descriptive information on a single entity. In this chapter, adopting more "open" product development practices within a software R&D group in Alcatel-Lucent Bell Labs, Nozay, France is studied.

There are many company case studies which highlight successful implementation of "open" approaches for the development and the commercialization of technology. These studies are generally treated within the *"Open Innovation"* literature (Chesbrough, 2003a). Mostly, these studies present successful implementations of open innovation and their results. The main unit of analysis in this stream of research is the firm. The goal of this chapter is to understand difficulties in adopting open innovation by a software R&D group within an incumbent firm with a substantial patent estate.

According to Chesbrough (2003a), many large firms not only have difficulties to profit from their vast knowledge stock but also fail to profit from actors outside of the firm boundaries to create and to market ideas. In order to overcome these problems, Chesbrough (2003a) proposes more open business strategies and firm organizations. Open innovation describes R&D as an open system into which different entities can reach and bring external sources of innovation to capture and create value. Business modeling is a fundamental element of open innovation. Defined in a single sentence; "open innovation is the use of purposive inflows and outflows of knowledge to accelerate internal innovation, and expand the markets for external use of innovation, respectively."(Chesbrough, 2006).

The level of analysis in open innovation literature is based on incumbent firms, however, according to Vanhaverbeke (2006) different levels of analysis are needed to understand and broaden the scope of open innovation. Moreover, case studies within this research stream are generally about the successful implementation of open innovation presented as a new innovation paradigm by Chesbrough (2006). Open innovation embraces many practices observed in FLOSS development model, and in addition it comprises innovative activities which use intellectual property rights such as patents and licenses. However, as presented in the previous chapter, software development and patenting are not complementary and could become contradictory in many cases. The case study presented in this chapter is concerned with the difficulties of a software R&D group which aims to adopt open innovation. The parent company has developed important patent portfolios over the years. Consequently there is an incomplete picture on difficulties of open innovation adoption for software development within an incumbent ICT firm. The research question which guided this chapter is; What are the barriers to adopt open innovation in an ICT company?

Implementing a new managerial concept is not an easy task. This chapter attempts to identify barriers of open innovation within an ICT firm. This process is discussed within the framework of the literature on routines

(Nelson and Winter, 1982) and "interpretive schemes" (difference in inter-organizational thinking and collaboration styles between different departments in large firms) (Dougherty, 1992). As far as managerial implications are concerned, findings of this research and discussions are expected to assist decision makers in their approach to the adoption of open innovation strategies.

In this chapter the next section presents the conceptual background on routines and open innovation. Then Section 7.2 will provide the methods used in this case study and the Section 7.3 will present the case. The following section will discuss the results and the last section concludes.

7.1 Conceptual background

In this section, two concepts are presented. The first concept is the organizational routines which is used in the interpretation and in the discussion of the results. The second one is the open innovation which is the new innovation strategy that the software R&D group aimed to adopt.

7.1.1 Organizational routines

"Routine" is a key concept in the evolutionary framework of Nelson and Winter (1982). According to Becker (2004), routines are the central unit of evolutionary analysis of firms. Routines are used to study different observations made on the operation of firms (Cohen and Bacdayan, 1994; Becker, 2004). Routines are regular and predictable patterns which are related to business behavior. It is argued that "well-defined routines structure a large part of organizational functioning at any particular time" (Nelson and Winter, 1982, p.97). Not all action patterns can be called routine, heuristics and rule of thumb found in internal functions of firms are put aside of the concept of routine (Cohen and Bacdayan, 1994). Pentland and Feldman (2005) argue that organizational routines have their internal structures and dynamics. Authors argue that, in order to analyze core organizational phenomena the internal structure of routines should be examined.

Routines are hard to observe and they are based on past experience (Cohen and Bacdayan, 1994). It is shown that routines are tacit and their imitation is difficult because individuals store organizational routines that they are responsible in their procedural memory. The procedural memory stored in individuals are the basis of the organizational memory. Consequently, "organizations remember by doing and organizational knowledge reside in its memory" (Nelson and Winter, 1982, p.99)

Routines are context dependent, even if they fail in some instances, routines should be measured by what they achieve. When the context is related with motivation, power relations and intra-organizational conflicts within organization comes to the surface. The idea of "routines as truce" could help to examine this situation (Nelson and Winter, 1982, p.107). Routines help to settle "divergence among organization members". However, Cohen and Bacdayan (1994) warn that motivational issues cannot be solved with the routine as truce formulation if there is a broad temporal and geographical scale.

Routines are difficult to change. They are specific to context thus it is hard to transfer to another context. Difficulties with respect to the transfer of routines are due largely to the tacit elements that they contain (Nonaka, 1994). There is not any one size fits all best practices, "there can only be local best solutions" (Becker, 2004). Moreover, routines are not static, they can evolve with the technological and industrial change (Pentland and Feldman, 2005).

Cohen et al. (1996) reached an agreement to define routine as "an executable *capability* for repeated performance in some *context* that has been *learned* by an organization in response to *selective pressures*".

7.1.2 Open innovation

The open innovation framework refers to the idea that by using both external and internal knowledge it is possible to accelerate innovation (Chesbrough, 2003b). Firms, which have successfully adopted open innovation, create internal mechanisms to access external knowledge and ideas and design business models which create value through their combinations. In open innovation, business models are at the core of both value creation and value capture strategies. Moreover, open innovation is also a tool which would create a market for the internal knowledge not used in the current business of the firm. There are various channels to market this internal knowledge of the firm. One of the important tools which are used in open innovation process is the intellectual property rights management (Chesbrough et al., 2006). The well known schema representing the open innovation is given in Figure 7.1.

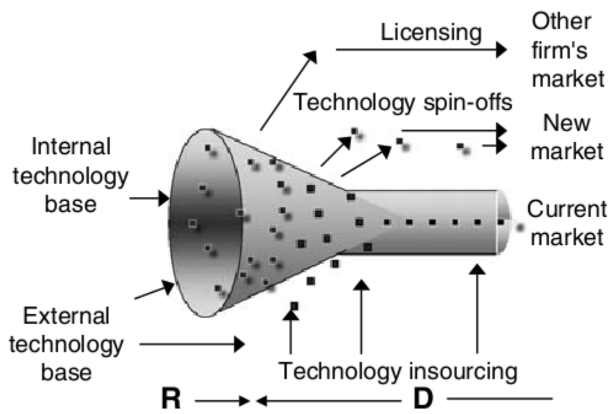


Figure 7.1: Open innovation (Chesbrough et al., 2006).

Open innovation marks a shift in innovation paradigms from closed to open models. Examples are obtained mainly from ICT firms such as Xerox, IBM, Intel and Lucent which have all dedicated chapters in the first book on open innovation written by Chesbrough (2003a). According to Chesbrough, these firms have recognized later or earlier that there is abundant knowledge outside of the firm boundaries. During the twentieth century, centralized R&D organizations in large firms are weakened. During this period, the quantity and in the quality of research centers have increased and universities played an important role in the dissemination of knowledge in various fields. Knowledge became more distributed and easy to access. Combined with low-cost Internet, it became possible to have access to an abundant knowledge far more easily than before. These changes during the twentieth century forced firms to search knowledge outside of their firm boundaries to create and to capture value in collaboration with different actors (Chesbrough, 2003a).

The closed innovation model is based on the idea that the firm which manage all innovative activities internally would be the first ones to enter into the market. Moreover, with the support of a restrictive IP management other firms would be prevented to do incremental innovations. Firms which have adopted closed innovation models often fall into the wrong idea that hiring the smartest people is the key to competitive advantage. The closed innovation model was adopted by most of the US major corporations during the twentieth century (Chesbrough, 2003a). This innovation model is represented in Figure 7.2.

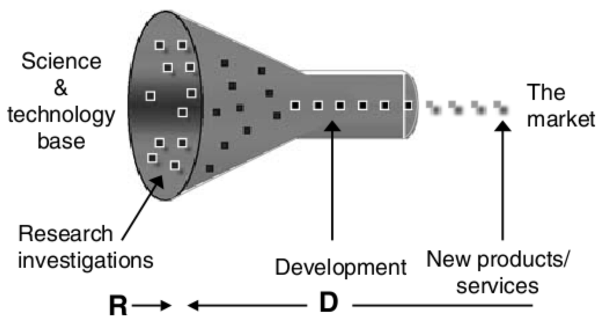


Figure 7.2: Closed innovation (Chesbrough et al., 2006).

Among various external sources of knowledge such as customers, suppliers, competitors, public and commercial research institutes, and databases, there are few interesting examples which appear in many open innovation articles. These examples are about the innovation intermediaries using Internet and crowd sourcing (Enkel et al., 2009; Albors et al., 2008). These sites join unresolved problems of R&D departments with experts scattered around the globe. Sites that are studied are Procter and Gamble's "Connect and Develop" (Dodgson et al., 2006) and InnoCentive(Lakhani and Lonstein, 2011)[1].

One of the pillars of the open innovation is its extensive use of the IP management tools such as patents and licensing. Proponents of a strong

[1]Other similar sites are yet2.com and NineSigma.

IPR system claim that there is an important amount of underutilized patent assets which could be used as an important tool within IP management strategies. However, the counter argument is that there is a little number of patents which have considerable worth. Evaluating the value of a patent is very difficult, especially without a sound business model supporting invention. For this reason, open innovation framework recommends that firms should also develop business models supporting their patents. In this way, licensing value of a technology would become clear for both licensee and the patentee. Various cases for different technology groups are given by Chesbrough (2003a).

Within the open innovation literature IPR management is one of the main tools to generate revenue (Simcoe, 2006). Standards play an important role in innovation process since they set the rules that govern the interfaces, which enable the interoperability of independently designed and manufactured products. An accepted standard can determine the survival of the companies and the dominance of a technology (Shapiro and Varian, 1999; Suárez, 2004).

There is a trade-off between open standards and closed standards. If the proprietary standards are adopted first entrants into the market have higher chances to capture a larger share of the market. However, open standards encourages new entries creating a perfect competition for all firms active in this area. Developing a product within a patented technology is risky for firms due to patent litigation cases. Moreover, close standards might also push manufacturers to develop new standards resulting with incompatible products with respect to the accepted ones. An open standard setting example such as FRAND, which groups many standards consortia, encourages its members to make their patents available to each other under fair, reasonable and nondiscriminatory licenses (Leiponen, 2008). And as the FLOSS gains an important momentum, product development and open standard setting with the participation of various entities became an example for many industries for its positive economic effects (Ghosh, 2005).

There are also many critical views about open innovation framework. A debate on open innovation started in the Technovation Journal questioning its novelty (Groen and Linton, 2010). FLOSS with its decentralized, open and participatory innovating and software development model, distributed with various licensing schemes have inspired the open innovation literature. However, FLOSS movement also has a negative connotation on strong IPR regimes, which is argued not only to be detrimental to FLOSS (Stallman, 2002) but also to the software technologies (Nelson, 1994; Mazzoleni and Nelson, 1998b). Discussion on this issue is given in Chapter 6. There is also a discussion on the terminology, von Hippel (2010) points the competing meaning of the term "open" as it is used in open source software.

The popularity of the open innovation concept resulted in a number of literature reviews on the subject. Dahlander and Gann (2010) review 150 papers and categorize the existing literature in four different forms of openness; revealing (outbound innovation and non-pecuniary), selling (outbound innovation and pecuniary), sourcing (inbound innovation and non-pecuniary) and acquiring(inbound innovation and pecuniary). Schroll and Mild (2012) review 282 documents with a focus on large scale quantitative-oriented studies. They analyze the literature on various components of the open innovation adoption measurements. Su and Lee (2012) analyze 130 open innovation research papers. This last article visualizes the research community structure with network analysis tools. It also provides important or emerging open innovation components.

7.2 Method

This case study was carried out during 2010 and 2011. The main methodology used in this case study is participant observation and open ended interviews (Yin, 2003). Interviews were conducted to understand how the video indexing technologies were developed, which was the subject of Chapter 4 and 5, they were also aimed to understand how open innovation practices were embraced by this research group. Articles and presentation on several projects made by the research group are also reviewed in the R&D building. Other than technical issues regarding video indexing technology, all interviews followed a structure; how a project is developed, what are the inflow and outflow of knowledge, how business partners are found, and what are the attitude of the members of the research group toward free and open source software and open standards. Open ended interviews are conducted with the R&D project manager of the video indexing technology and several engineers working for the same project. Moreover, research engineers and project coordinator within the academy are also interviewed. Interview questions to the R&D group members are given in the Appendix C.

Several meetings also took place during 2010, 2011 and 2012 with different researchers who collaborated with the R&D team. During the case study the R&D laboratory was visited once a week for four months then once every two weeks for another three months.

7.3 Background

Results are analyzed and presented by relying to the theoretical propositions of the open innovation. The most important important aspects of the open innovation is the inflow and outflow of knowledge through firm boundaries. Moreover, IPR management and inter-departmental relations are also very crucial in the application of open innovation. In this sections results obtained are presented within this four aspects of open innovation.

7.3.1 Brief information

Alcatel-Lucent is a global ICT equipment company which holds Bell-Labs, one of the largest R&D laboratories in the ICT industry. Alcatel merged with Lucent Technologies in 2006. The patent portfolio of Alcatel-Lucent comprises over 30,700 patents[2]. The case study is carried out within a software R&D research team in Alcatel-Lucent Bell Labs, Nozay, France. Compared to the whole worldwide R&D staff, the software application R&D team, with which the case study is conducted, represents only a small fraction with around 60 researchers.

[2]http://www3.alcatel-lucent.com/wps/portal/AboutUs/factsheet

The software R&D team has the duty to work and develop applications which takes into account the ubiquity of computers. The R&D team works on different projects and the number of researchers can vary according to the needs of the project. Team leaders are trying to minimize the number of projects that one researcher is working on in order not to distract researchers. Even though the interviewed people pointed out that there is no time constraints on research projects they also emphasized that in fact the process is more complicated.

One of the important projects of the R&D group is the development of a platform for multimedia content indexing with search capabilities. The platform is designed such that it can use numerous content analysis modules which would work on top of the platform. However, not all modules are intended to be developed within the company because the number of required modules is very high and various competencies in different areas are needed.

7.3.2 Inflow of knowledge through external partnerships

Historically the research laboratory has developed many important materials, computing devices and software. However, among R&D members a "not invented here syndrome", which is discussed within the open innovation literature, could not be observed (Chesbrough and Crowther, 2006). On the contrary, it is found out that members of the R&D team are open to research collaboration with partners in the industry as well as in the universities to obtain the missing modules. Researchers are also keen on increasing the number of research collaborations with R&D groups of universities. One prospective research partner adopted FLOSS development model and the R&D team was keen on using the same licensing scheme. To summarize their approach on obtaining knowledge outside of the firm boundaries, one of the R&D team leaders said that *"it would be stupid to redo things which are already done."*

Interviewed people stated that they follow and are aware of the research conducted by incumbent firms within their industry. However, they also recognize that the technologies by small start-ups can be unnoticed. Most often, these start-ups connect to the research laboratory and not the other way around. During the interviews it is found that some finalized research programs created industrial partnerships with firms found through informal networks of researchers or managers.

7.3.3 Outflow of knowledge

Free outflow of information from the software R&D group is limited. However, if the research results are mature enough, researchers have a positive approach to release some of the code developed within the R&D laboratory and share it with the FLOSS community. After the firm as a whole took the decision to opt open innovation practices, a web based hosting service for open source software projects developed within the company is also setup during the same period. After three years of activity, the project hosting service had only a dozen software projects and most of them do not have much activity[3].

The company is taking part in many standard setting activities for various technologies. Related to the video content analysis an interviewed person said, *"We are not trying to impose something, we try to be pragmatic. We use something that is as usable as possible with an industry acceptance"*. It was also pointed out that they prefer not to develop a closed standard on their own but could trim or mask some part of the large standard to make it more specific and easy to use.

7.3.4 IPR Management

During the interviews it was clear that the members of the R&D team aim to file patents which is one of the fundamental measures of their performance. Related to the patent analysis, one of the important discussion topic is whether the research priority should be set according to any patent analysis and forecast. The R&D management would like to set their research topics according to forecasting tools and methods which could be based on patent

[3]http://open-innovation.alcatel-lucent.com/softwaremap/full_list.php

analysis. On the other hand, all patent related activities are done only by the IPR management team. Unfortunately, in various meetings the IPR team did not reveal the patent analysis conducted within the company due to the culture of secrecy which is prevalent in many levels within the firm. Moreover, from the interviews with the researchers, it is understood that they do not have any access to technology forecasting results from IPR department. In addition, it is also not clear what kind of forecasting methods are used by the IPR department, if made. It appears that the IPR department is more involved with patent filing, patent licensing and especially patent litigation cases.

7.3.5 Inter-departmental relations

According to the researchers, the aim of the project group is to acquire capabilities in various domains and push the boundaries of the technical knowledge. However, it is expected by the management that before being accepted, research projects should be accompanied by business model. Researchers have some disdain to develop business models in addition to their research projects. In order to understand how market related information is acquired and to understand the position of the marketing department with the R&D laboratory a single meeting is organized with the marketing department. It is deduced that the R&D department is conducting research without any interaction with the marketing department. The marketing department enters into the scene only when the product is ready to be introduced to the market. The marketing department takes into account international market differences in its activities.

The final observation obtained from the software R&D group is that the R&D department, while contributing to the scientific community, has very limited interaction with marketing and IPR departments. It is also observed that the company follows a "linear" model of innovation. R&D department generally passes the concepts to engineering department which deals mainly with the productivity and effectiveness of the code. These two departments interact to release a product and the R&D department has a strong preference on setting their own research agenda. There is also an important bureaucracy within the firm which creates many insular departments having their own duties with predefined rules. However, the detachment of departments results with the pressure on researchers also to find an acceptable business model. The business model has a key role to obtain the approval for the research proposition.

7.4 Discussion

This case study presents interesting observations related to the open innovation literature. Bell Labs, the most important industrial research laboratory of the world at the times of closed innovation paradigm is known for having a “notoriously inwardly focused culture” (Chesbrough, 2006, p.2). Bell Labs had been broken up several times since 1980s after the AT&T monopoly case. Lucent which is established in 1996 acquired the last and the biggest portion of the Bell Labs. According to Chesbrough (2003a), Lucent’s New Ventures Group (NVG) has been an important and one of the first example of open innovation. Lucent’s NVG has been successful in leveraging technology from Bell Labs.

Even though Alcatel-Lucent endorsed open innovation, there is no study on the effects of merge on open innovation activities conducted within the teams inside the firm. In this exploratory case study, difficulties to the adoption of open innovation that a software R&D group within Alcatel-Lucent Bell Labs, Nozay, France encounters are investigated.

The methodology used in this case study is participant observation and open ended interviews (Yin, 2003). Barriers which are encountered in the adoption of open innovation by this R&D group is analyzed through the concept of routines (Nelson and Winter, 1982).

During the interviews, enthusiasm of the researchers for more open practices such as research collaboration with outside research groups, their use of FLOSS in some tasks are observed. Moreover, they did not show any “not invented here” syndrome even though the laboratory hold 12 Nobel prizes. These activities are inbound open innovation practices which are only looking to the mechanism to bring the outside knowledge into the innovation process of Alcatel-Lucent Bell Labs, Nozay, France. However, some difficulties related to outbound knowledge flow is observed. Moreover, difficulties to develop business models which is at the core of the open innovation, is also remarked.

Enkel et al. (2009) found out that barriers to invest and profit from open innovation initiatives are; loss of knowledge, higher coordination costs, loss of control, difficulty to find the right partner, and imbalance of open innovation and daily business. Alcatel-Lucent is within the league of global ICT equipment producers with Ericson, Huawei and Cisco. The core competency of Alcatel-Lucent is ICT network hardware. In this case study, difficulties of adopting open innovation by the software R&D group are related to the difference of hardware production dominance within Alcatel-Lucent and the detachment of various departments within the company which is related to the coordination cost.

As Cohen and Bacdayan (1994) argue, routines which are based on past experiences are difficult to observe. Various differences related to hardware and software production and the core competency of Alcatel-Lucent shape their routines which depend on hardware. Hardware and software IPR differs largely due to their technological nature. Software is argued to be a "cumulative system technology" (Nelson, 1994); new technology is build upon the interactions of existing technologies. However, technologies like hardware are protected under the traditional, discrete invention model (Nelson, 1994). A software product inevitably uses various patented inventions which are obtained by different producers but a hardware producer could circumvent other patents by developing new technologies and give a product which does not infringe others' IPR. The peculiarities of the software technology vis-à-vis IPR is dealt in various chapters of this thesis.

The main argument of the first part of the discussions is that the difference of hardware and software technologies should be taken into account when creating IPR strategies. However, in this case, a firm, which has set its IPR related routines mainly on hardware technologies, faces difficulties as revealed by a software R&D group in need of consistent open innovation strategies.

Routines and common beliefs are so much accepted by researchers that some of the people from the software R&D group argue that patents related to software and hardware do not differ.

In open innovation, value creation and value capture require coordination between departments. One of the surprising findings of this case study is the will to adopt open innovation among researchers while all intra-firm departments are detached from each other. This is mainly due to the innovation process of the firm as a whole which is found to be linear. The patenting and IPR divisions enter into the scene when the project is finished. The IPR division gets in touch when the research is found to be "valuable" enough to be patented. And the marketing division gets into contact with the engineering team when the product is ready to be launched. A tight collaboration with various departments inside the firm from the idea conception to marketing is needed. This is especially important so that researchers in R&D can design new business models which suit their software contribution. Perhaps, a second organizational unit is needed to implement open innovation and overcome barriers to the change.

Because of the weak coordination between departments, researchers face numerous tasks. These new tasks are patent and market analysis, searching for possible business partners, and business model development. As a result, a tension raises within the R&D teams on the conception of the research agenda. Should researchers also develop viable business models to obtain an approval from their superior to continue on their research?

It has been argued that closed and open innovation cultures are different due to the difference in levels of risk taking and openness to new ideas (Herzog, 2008). Results are more related with the truce between departments. Routines as truce, in the above example, which was attained between intra-organizational departments could be destabilize with the introduction of open innovation. Adoption of the open innovation is pushing marketing and IPR departments to be involved with new tasks. However, adoption and application of open innovation practices were not endorsed by these departments. And, there are also some findings in the literature on routines which show that the increased pressure will lead to a preference of the most repeated routine (Becker, 2004). In that case, it might be difficult to adopt open innovation so that various departments collaborate more densely.

In this case study it is observed that the detached operation of the departments of Alcatel-Lucent working in different "interpretive schemes" of innovation (Dougherty, 1992). It is argued that departments have different way of thinking and collaborating, which forms an important barriers to open innovation. Dougherty (1992) argues that there are two interpretive schemes which create interpretive barriers: 1) divergent "thought world" of different departments 2) routines which are separating rather than coordinating these thought worlds. It would be simplistic to consider a structural reorganization to ease the adoption of open innovation.

There are several managerial implication of this study. In order to implement a new innovation strategy such as the open innovation, it is important and difficult to make various changes within the firm. Two important findings should be taken into account on changing the firm innovation strategy into open innovation model. These are 1) organizational detachments of different intra-firm entities and 2) different thought worlds inside the enterprise. Managers should take into account these two factors for the success of the adoption of the open innovation.

7.5 Conclusion

In this case study the adoption of open innovation by a software R&D group at Alcatel-Lucent Bell Labs, Nozay France is investigated. Other than the R&D group, two other departments within the company, which have important roles in open innovation, are also taken into account. It is found out that the IPR department is secretive and appears not to make any technological forecast based on patent analysis. Marketing department is focused only on marketing products without providing any feedback to the R&D group. The studied R&D group is avid to adopt open innovation. These three different groups represent three different "thought worlds" which are shaped by different routines. In addition to the conceptualization made by Dougherty (1992), it is possible to add that to the causes of different interpretive schemes between the whole and the software R&D group is the technological difference and its relation to the different IPR schemes. The core competency of the firm is network hardware design and production. This technological difference is also increasing the separation between departments and also lessens the coordination of the observed software R&D group with the others.

Taking the example of Dougherty (1992), interpretive barriers could be overcome with cultural solutions which could start by recognizing the difference between software and hardware in IPR management and business models which are at the core of the open innovation. Moreover, increased communication between various departments from the start of the R&D process would bring successful adoption of open innovation.

Chapter 8

Conclusion

ICTs play a considerable role in economic development (Freeman, 2007) and software technologies are recognized as significant drivers of economic growth (Lippoldt and Stryszowski, 2009). Software technologies have important differences compared to other industrial artifacts. Information is the main input for the software industry; it is inexpensively reproducible after the first copy and it is also inexhaustible (Steinmueller, 2007). In this thesis the structural and institutional aspects of the software technologies are investigated. These aspects have substantial differences compared to the technologies which have manufactured products as output. The structural characteristics of the software technologies which are treated in this thesis are; *economies of scope* and *modularity*. The institutional peculiarity of the software technologies is the *intellectual property rights* regimes.

The modularity of the software and its deployment capacity for various market and purposes create economies of scope. Moreover, IPR regimes which have evolved considering the manufactured goods create an eco-system in which software technologies face various barriers for innovation. The characteristics of software technologies also gave rise to different business models. Taking into account these characteristics of the software technologies, the innovativeness in the software industry and the effect of various IPR regimes on software are still subject to debate. This thesis aims to contribute to these discussions by understanding how the innovativeness in software could be promoted and how firms could balance different software development models.

The main research questions treated in this thesis are as follows;

- What are the managerial implications to promote innovation in software sector as far as the IPR regime is concerned?
- How should firms balance FLOSS and proprietary development models for innovative performance?

To answer these broad questions, first the concept of key factor of the fifth technological revolution, age of information and telecommunication is discussed within the techno-economic paradigm literature (Freeman and Perez, 1988). In this thesis, different methodologies are used to answer the above research questions and several software technologies are considered to illustrate the research questions. Economies of scope are investigated by using an agent based simulation. The research on modularity is carried out exploiting a patent analysis of the video indexing technology. The IPR issue is examined within two different chapters. In the first one, using aggregate data on firms contributing to the Linux kernel project a panel data analysis is conducted. It is aimed to understand the effect of patenting and contribution to the Linux kernel project on the performance of firms. The second one focuses on a case where open innovation is aimed to be implemented by a software R&D group within Alcatel-Lucent Bell Labs, Nozay, France. These unique examples cannot give any macro trend on software industry but they could provide some insights to address the above research questions.

The following sections will present the results obtained from the chapters which treat the concept of key factor and the three aspects of software technologies. Then some concluding remarks will follow, to discuss the managerial implications of the results obtained.

8.1 FLOSS as a key factor

Software technologies are built and developed incrementally; they are based on combination and reconstruction of inputs (Hall and MacGarvie, 2010).

However, there are two opposite models in software production. The first one is the proprietary software development model with which software products are introduced into the market without source code. These software are generally developed within a hierarchical and opaque organizations. On the other hand, the second mode, FLOSS is developed with the contribution of developers distributed around the globe. The source code of FLOSS is available to anyone connected to the Internet. This model provides various licensing schemes at different "openness" levels. Contrary to the FLOSS licenses, proprietary software licenses usually have many restrictive clauses. The controversy around the two opposite software development models is reflected in the title of the book "*Cathedral and Bazaar*" (Raymond, 1999) which makes an allusion to this dichotomy in software production. However, in between these two extremes, there are numerous business models, licenses and development models which are the combination of these two opposite models. This combinations are mainly related to the IPR issues which also affects the organization and product development.

In Chapter 2 the concept of key factor in the fifth techno-economic paradigm is discussed. Key factor is the main resource behind each techno-economic paradigm and according to the definition given by Freeman and Perez (1988), it has three properties. The key factor should have (1) relatively low and rapidly falling costs, (2) unlimited supply over a long period of time and (3) possibilities to be used in a large number of products and process through out the economy. The key factor is also accepted as one of the contributors to the creation of a new social structure.

FLOSS gives its users the freedom to "run, copy, distribute, study, change and improve the software" through its license (Stallman, 2002). The cost to obtain the source code is negligible and as long as the project is active, the source code generally evolves into a better version if there is an interest within the FLOSS community to develop and to use it. FLOSS is used in many computerized products and it runs behind various services. There is an increasing number of examples that the development model of FLOSS is transposed to other industries (Lerner and Tirole, 2002). FLOSS development model and licensing have influenced the production and licensing of other copyrighted materials such as image, music and movies. Open innovation as a management model is also inspired by the FLOSS development model and its way of using knowledge (Chesbrough et al., 2006). In Chapter 2, it is argued that FLOSS fulfills the definition of the key factor.

8.2 Economies of scope

A software with small changes can be used in different products targeting various markets. It is a common practice to make alliances among software firms. These alliances are based upon the complementarities of resources of involved parties (Hagedoorn, 1993). These alliances are between firms which are active in different markets or which have different knowledge bases. There are also examples of alliances of firms with limited software development capabilities. The number of alliances among software firms have been steadily increasing since 1990 and it has reached to cover 95% of the software firms in 2000 (Lavie, 2007).

In Chapter 3, Saviotti and Metcalfe (1984) model of innovation is the starting point in investigating economies of scope in software industry. This model not only allows technological distance (overlap) but also helps to introduce market distance between firms. As a result, this model helps to understand the learning dynamics of firms under different IPR regimes and knowledge codification settings.

The research questions of this chapter are; 1) what are the dynamics of learning in a system with rich economies of scope? 2) To what extent does learning depend on IPR regime and knowledge codification?

An agent based simulation is conducted to answer the above research questions. The simulations start with firms scattered randomly and uniformly on a torus shaped surface defined by technology and market dimensions. To create heterogeneity among the firms, their partner selection preferences are set randomly based on market and technological distances with their possible partner. Simulations are carried out for different IPR regimes and knowledge codification schemes. Results are analyzed with respect to partner selection criteria of firms.

The results reveal that the cooperation of two different firms which are close in one dimension (technology or market) and distant in the other, the ambidextrous ones, bring the maximum number of alliances and the highest accumulated knowledge. Firms which prefer to make alliance with the firms having a similar alliance preference, the homophilic ones, procure the least learning. As economies of scope refer to a similar knowledge being applied in different markets, this chapter underlies that, software firms can leverage this by forming alliances with partners from different industries.

8.3 Modularity

Modularity is a concept that helps to overcome the increasing complexity of technology (Baldwin and Clark, 1997). Most often, modularity is studied under product architecture (Ulrich, 1995). On the other hand, modularity in software has been an important research subject, where the leading work belongs to Parnas (1972). Modularity in software aims to develop reliable products in short periods of time and to increase the number of reusable software code.

There is an important discussion within the modularity theory which asserts that technological modularity has also an effect upon organizational structure (Sanchez and Mahoney, 1996). It is argued that modular product systems can be accompanied by increased specialization at the firm level (Langlois and Robertson, 1992) or it can result in the emergence of diversified firms.

There are two opposing views on the relationship between specialization/diversification and innovative competences of firms. The first one argues that through cross-fertilization of ideas, diversified firms are more innovative. The second claim stresses that firms doing R&D in few areas could be more successful in innovativeness (Brusoni et al., 2001). Chapter 4 and 5 aim to answer whether firms making important technological achievements to the development of a software technology are specialized or diversified and how this situation changes throughout time.

Research questions addressed in Chapter 4 and 5 are; 1) do technological achievements in a highly modular software technology originate from the firms which are diverse or which are specialized? 2) What are the average knowledge boundaries of firms contributing to the development of a software technology?

The two chapters, Chapter 4 and 5 complement each other. Chapter 4 presents the theoretical background on three main concepts used in Chapter 5. These concepts are modularity, patent analysis and technological diversification of firms.

In Chapter 5, video indexing technology is considered to address the above research questions. Video indexing technology is used to automatically extract semantic information from video files. For this analysis, first the modules of video indexing are identified, then the product system architecture (Ozman, 2011) of the video indexing technology is set. Afterwards, these modules are grouped according to the framework given by Snoek and Worring (2005) that decomposes video. Modules are grouped under visual analysis, audio/speech analysis and optical character recognition technologies.

For each of the three modules and for video indexing, patent connectivity analysis is conducted. This method is developed by Hummon and Doreian (1989) and improved by Verspagen (2007). From the connectivity analysis, knowledge flow maps are obtained. From these maps, patents and firms which made significant technological achievements over the period 1970 to 2010 are obtained. Finally the technological diversity of the firms, which are determined from previous analysis, is measured with respect to their patent pool.

It is found that firms in video indexing and its modules do not evolve into an increasing specialization as Langlois and Robertson (1992) suggest. Moreover, firms which have made technological achievements are found to be highly diversified. Furthermore, there are few firms which are highly specialized, and have also made technological achievements in the development of different modules.

8.4 Intellectual property rights regimes

IPR regimes are the institutional features of software technologies. Since the dawn of the computer era, IPR has a considerable part in the development of the software industry. One of the first spin-offs in the history of computer is created in 1946 by two researchers who left the ENIAC project due to a patent dispute between researchers and the university in which they were working (Ceruzzi, 2003). Compared to the previous technologies and to the computer hardware technologies, IPR issues became more complex with the development of the software technologies and institutional changes which have shaped the acceptance of software patents since 1970s (Hall, 2004).

To promote innovation in the software sector a balanced approach to the IPR regimes is advocated in an OECD report (Lippoldt and Stryszowski, 2009). In this report it is recognized that software industry needs a fine balance between rewards for innovation incentives such as patents and licenses and the collaboration of its users and developers who are outside of the firm boundaries. The proposed solution is the creation of an institutional setting which would not hinder collaboration and co-development.

A balanced approach is likely to take some time to be set and this thesis aims to contribute to this debate. Firms have already developed a variety of business models which mix proprietary and FLOSS technologies and solutions (Bonaccorsi et al., 2006; Henkel and Baldwin, 2009).

The hybrid business model is a synthesis of proprietary and FLOSS product and service development. Firms adopting hybrid business models use this model for an easy entry in to the market. It is also observed that firms using this business model do not evolve into pure FLOSS or pure proprietary software and service development (Bonaccorsi et al., 2006). IP modularity (Henkel and Baldwin, 2009) has some similarities with hybrid business model. The concept of IP modularity is the release of some source code open to the community to contribute and co-create but at the same time to retain most of the important part of the software closed and proprietary.

In chapters 6 and 7, the balance of more "open" software development models and restrictive practices such as patenting is investigated. First, in

Chapter 6, using aggregated data, contribution to the Linux kernel project and the effect of patenting on performance of firms is investigated. For this purpose a panel data analysis is conducted on data comprising 169 firms which have contributed to the Linux kernel project between 2005-2011. Second, in Chapter 7, a case study on the barriers to adopt open innovation by a software R&D group in Alcatel-Lucent Bell Labs, Nozay, France is presented.

Results obtained in Chapter 6 show that patenting and contributing to the Linux kernel project are positively correlated with sales. However, their interaction is negatively correlated, it is found out that patenting and contribution to the Linux project are substitutes. From these results it is argued that firms are locked in an incompatible hybrid model. Moreover, firms give away some of their inventions for free by the expectation that the community would overtake the development of the given contributions.

In order to have more insight on the effects of IPR on software development a case study is conducted. In Chapter 7, difficulties in adopting open innovation by a software R&D group in Alcatel-Lucent Bell Labs at Nozay France are studied. The methodology used in this case study is participant observation and open-ended interviews (Yin, 2003). Results are discussed by referring to the concepts of routines (Nelson and Winter, 1982) and "interpretive schemes" (Dougherty, 1992).

The adoption of a new innovation model is a tedious process. In this case study various barriers to the adoption of open innovation related to software are highlighted. The first observation is the lack of separate IPR strategies for hardware and software technologies. In other words, the case study highlights the dominance of the IPR management routines based on hardware technologies which are not suitable for software products. The second main finding is the detachment of IPR and marketing departments with the software R&D group. These three groups have different "*thought world*" (Dougherty, 1992) which are shaped by different routines. These routines have served not only to create a truce between departments but it also created a detachment among department of the firm. It is claimed that the adoption of open innovation requires an inter-departmental collaboration.

In this chapter it is argued that recognizing the difference between hardware and software IPR management and business model could decrease the interpretive barriers and ease the adoption of open innovation.

8.5 Conclusion and managerial implications

This thesis investigated the structural and institutional aspects of software technologies and aimed to determine managerial implications to promote innovation in software sector in relation to the IPR regimes. Moreover, it explores the balance of the FLOSS and proprietary software development models at firm level. For these purposes structural aspects of software technologies which are economies of scope and modularity are also taken into account.

In Chapter 3, it is showed that ambidextrous firms create more alliances with firms which complement them in market or in technological dimensions. And it is also found that learning is higher for these firms having alliance preferences with partners close in one dimension and distant in the other dimension. In line with the concept of diversity in evolutionary economics, as a managerial implication, this simulation shows that software firms which are in rich economies of scope should have access to better interfaces to create partnerships with firms active in different markets and technologies.

Modular characteristic of software technologies is well suited for product differentiation in rich economies of scope. The results obtained in Chapter 5 show that in video indexing and in its different modules, firms are technologically very diversified and are not specialized software firms. Moreover, there is no sign for specialization on a time-line. Furthermore, in some cases a vertical innovation is also observed for some firms. Results of these two chapters support the argument that knowledge diversification is an important asset to innovate and due to modularity there is not much sign for technological specialization of firms.

Software developed by technologically diversified firms create a system in which the development of various business models are required in order to open new markets. However, as it is shown in Chapter 7, to achieve the adoption of more "open" innovation practices and the development of new business models an “interpretive cohesiveness” among various department of the firm is needed. Without this cohesion, intra-firm cooperation, which is needed to find new markets and new business models, would be difficult to obtain.

In Chapter 3, simulations showed that within tighter IPR regimes firms learn less and the number of alliances decreases. Moreover, it is found that firms become an easy target for non-practicing entities within a tighter IPR regime and the increasing number of patent litigation cases will hinder the innovativeness of firms. Taking into account results obtained in Chapter 3, the balance of proprietary versus FLOSS code is on the side of the FLOSS code which has the advantages to be easily attainable by all size of firms targeting numerous markets. FLOSS is an important technological choice within a rich economies of scope. However, as it is also documented within the case study in Chapter 7, adoption of more “open” innovation strategies could face various difficulties.

The balance between FLOSS and patenting is found to be difficult to attain in Chapter 7. Moreover, in Chapter 3 it is shown that strong IPR regimes are not for the advantage of firms but also for the whole system as the learning decreases. The effect of a strong patent system on the economy and the innovation is still not clearly set (Mazzoleni and Nelson, 1998a). Based on examples from nineteenth century, patenting is not really a pre-requisite which incite innovation (Moser, 2003, 2007). It is shown that several countries’ patent laws were rudimentary, weak or in the case of the Netherlands, it was abolished. The lack of patent system did not decrease the innovativeness of states. One of the key arguments to abolish patents is based on the idea of the free enterprise (Machlup and Penrose, 1950). Fervent discussions between opponents and proponents of the patent system, in the nineteenth century, used nearly all arguments which are still valid today. Today’s discussions revolve around four theories; (1) motivations of patents to invent, (2) encouragement of the use of patented invention, (3) patents induce development and commercialization of inventions and (4) patents enable orderly development of broad prospects (Mazzoleni and Nelson, 1998b,a).

It is known that all inventors do not prefer to patent all their inventions

(Mansfield, 1986). Based on data from four world's fair between 1851 and 1951, it is argued that inventors do not patent if they can rely on secrecy but, if it is possible to reverse-engineer then inventors prefer to patent (Moser, 2007). It is also argued that inventors prefer secrecy to patenting (Levin et al., 1987). However, keeping secret software technologies for non-trivial software systems is difficult. If computing time allows software could be reverse engineered and its code and used algorithms could be understood. Keeping the source code secret or patenting limit the open interaction between software users and producers.

Reducing the life time of software patents would push incumbents to adopt more "open" business models which do not rely solely on licensing. Moreover, if the patent life time of software is reduced, firms would be encouraged to use collective invention and collaboration. Beside the difficulties to define a software patent, a possible change in software patenting would promote innovation in this highly modular system. However, institutional factors are very important in these discussions.

A faster innovation pace in software requires an acceptance of open standards and respect of privacy and security of users which were not taken into account in this thesis. Open standards are defined as standards which would allow equal access for firms which would implement them. Open standards would allow the development of natural monopolies while at the same time ensure full competition among suppliers of that particular technology (Ghosh, 2005). Open standards diminish lock-ins and reduce concentration, while at the same time they increase firms specialization (Malerba et al., 2008). The use of proprietary standards became infeasible for IT vendors due to various technical and economic reasons. These vendors embraced open standards due to their low market share to sustain proprietary R&D, difficulties to stand against demands for open standards, problems in establishing a proprietary standard, and the willingness to shift competitive advantage to another architectural layer (West, 2003). Open standards lower the entry barrier and incite imitation.

According to the OECD report on the Innovation in the Software Sector (Lippoldt and Stryszowski, 2009), participation of end-users into the innovation process, security and balanced IPR regimes are some of the important concepts which would contribute to the innovation in software industry. FLOSS adoption and its application create new markets especially where security is needed. The Internet economy relies deeply on security and privacy. FLOSS development model brings transparency with the participation of users as well as developers. Moreover, FLOSS development model also leads to code review process which is open to anyone, thus independent software developers and security researchers can find and show security risks of the software. This development model also facilitates the discovery of computer bugs or security flaws which are patched by the fastest developer who get recognition for his/her contribution. On the other hand, in proprietary software development "security by obscurity" is the adopted security model and software users are generally at the mercy of the software developing firms. Even though the FLOSS development model can give a false sense of security, practice showed that it increases trustworthiness (Hansen, 2002).

FLOSS amplifies the characteristics of the software which are economies of scope, modularity and create an important discussion forum related to various IPR regimes. The adoption and use of FLOSS and more "open" business models have the power to push the knowledge boundaries of firms and increase the pace of innovation. But what if both opposite types of software development prevails? Through game theory, it is argued that a winners take all competition may arise and one of the software type looses the market. Firm can opt low price-high quality software or high price-low quality strategy. In any case credible FLOSS improves the utility of users (Darmon et al., 2007).

This thesis while answering some questions is also raising new research questions. In the second chapter FLOSS is argued to be a key factor of the fifth long-wave. Following this line of reasoning, it could be asked how this key factor will prevail or affect the following waves. Is FLOSS a key factor or a mediator for an open access to knowledge and development of new technologies? The simulation which is carried out in Chapter 3 could be tested with the addition of more parameters. An analysis of the software firms are carried out based on their patent portfolios in Chapters 4 and 5.

However, firm mergers and acquisitions and intra-firm modularity was not possible to be obtained through the patent analysis made in these chapters. A more refined intra-firm modularity analysis is required. In Chapter 6 a fine balance of two opposing IPR strategies are investigated, this work require a more refinement by investigating the characteristics and sectors of firms contributing to the Linux kernel project. The case study on the barriers to adopt open innovation also requires to test the results of the Chapter 7 in other software companies as well as in other industries.

Some managerial implications of these results are as follows: First, as far as alliance strategies are concerned, firms should exploit their existing competences by forming alliances with others operating in different markets. At the firm level, this not only results in higher learning but also helps knowledge diffusion in the industry which is one of the drivers of innovation (Ch. 3). Furthermore, the results reveal that diversified firms have a critical role in the development of proprietary software technologies (Ch. 5). Nevertheless, they face problems in shifting to a more open innovation environment. One of the reasons behind these difficulties is related with their routines which are oriented towards hardware. Another difficulty is related with lack of coordination between different departments, like IPR, R&D and marketing (Ch. 7). Fostering knowledge exchange between departments is an important organizational strategy. This is especially important, since, proprietary software and FLOSS acts as substitutes in firms performance (Ch. 6). Designing and implementing strategies to foster open innovation would help to reduce the extent of this substitution effect and help firms to find the correct balance between proprietary software and FLOSS. These results are especially significant considering the highly modular structure of software.

References

Ahuja, G. and Lampert, C. M. (2001). Entrepreneurship in the large corporation: a longitudinal study of how established firms create breakthrough inventions. *Strategic Management Journal*, 22(6-7):521–543.

Albors, J., Ramos, J., and Hervas, J. (2008). New learning network paradigms: Communities of objectives, crowdsourcing, wikis and open source. *International Journal of Information Management*, 28(3):194–202.

Alcácer, J., Gittelman, M., and Sampat, B. (2009). Applicant and examiner citations in U.S. patents: An overview and analysis. *Research Policy*, 38(2):415–427.

Allison, J. R., Lemley, M. A., and Walker, J. (2009). Extreme Value or Trolls on Top? The Characteristics of the Most-Litigated Patents. *University of Pennsylvania Law Review*, 158(1):1–37.

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

Anton, J., Greene, H., and Yao, D. (2006). Policy implications of weak patent rights. In Jaffe, A. B., Lerner, J., and Stern, S., editors, *Innovation Policy and the Economy, Volume 6*, number August, pages 1–26. The MIT Press.

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

Arora, A. and Gambardella, A. (2006). *From Underdogs to Tigers: The Rise and Growth of the Software Industry in Brazil, China, India, Ireland, and Israel*. Oxford University Press.

Arthur, W. (1989). Competing technologies, increasing returns, and lock-in by historical events. *The Economic Journal*, 99(394):116–131.

Arthur, W. (1994). Inductive reasoning and bounded rationality. *The American Economic Review*, 82(2):406–411.

Aslett, M. (2012). Open Source License Data.

Axelrod, R. (2003). Advancing the Art of Simulation in the Social Sciences. *Japanese Journal for Mangement Information Systems*, pages 1–19.

Baldwin, C. Y. and Clark, K. B. (1997). Managing in an age of modularity. *Harvard Business Review*, 75(5):84–93.

Barberá-Tomás, D., Jiménez-Sáez, F., and Castelló-Molina, I. (2010). Mapping the importance of the real world: The validity of connectivity analysis of patent citations networks. *Research Policy*, 40(3):473–486.

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

Basberg, B. (1987). Patents and the measurement of technological change: a survey of the literature. *Research Policy*, 16(2):131–141.

Batagelj, V. (2003). Efficient algorithms for citation network analysis. *arXiv preprint cs/0309023*, pages 1–27.

Baum, J. A., Cowan, R., and Jonard, N. (2009). Network-independent partner selection and the evolution of innovation networks. *Management Science*, 56(11):2094–2110.

Becker, M. C. (2004). Organizational routines: a review of the literature. *Industrial and Corporate Change*, 13(4):643–678.

Bekkers, R. and Martinelli, A. (2012). Knowledge positions in high-tech markets: Trajectories, standards, strategies and true innovators. *Technological Forecasting and Social Change*, 79(7):1192–1216.

Benzeghiba, M., Demori, R., Deroo, O., Dupont, S., Erbes, T., Jouvet, D., Fissore, L., Laface, P., Mertins, a., and Ris, C. (2007). Automatic speech recognition and speech variability: A review. *Speech Communication*, 49(10-11):763–786.

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

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

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

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

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

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

Breschi, S., Lissoni, F., and Malerba, F. (2003). Knowledge-relatedness in firm technological diversification. *Research Policy*, 32(1):69–87.

Breschi, S., Malerba, F., and Orsenigo, L. (2000). Technological Regimes and Schumpeterian Patterns of Innovation. *The Economic Journal*, 110(463):388–410.

Bresnahan, T. F. and Trajtenberg, M. (1995). General purpose technologies 'Engines of growth'? *Journal of Econometrics*, 65:83–108.

Breusch, T. and Pagan, R. (1979). A simple test for heteroskedasticity and random coefficient variation. *Econometrica*, 50:987–1007.

Brooks, F. (1987). No silver bullet: Essence and accidents of software engineering. *IEEE Computer*, 10(4):10–19.

Brusoni, S. and Fontana, R. (2005). Modularity as an Entry Strategy: The invasion of new niches in the LAN equipment industry. *KITeS Working Papers*, WP n.171(July):1–47.

Brusoni, S., Prencipe, A., and Pavitt, K. (2001). Knowledge specialization, organizational coupling, and the boundaries of the firm: Why do firms know more than they make? *Administrative Science Quarterly*, 46(4):597–621.

Bughin, J., Corb, L., Manyika, J., Nottebohm, O., Chui, M., de Muller Barbar, B., and Said, R. (2011). The impact of Internet technologies: Search. Technical Report July, McKinsey&Company.

Cameron, A. and K.Trivedi, P. (2005). *Microeconometrics : Methods and Applications*. Cambridge University Press.

Campbell-Kelly, M. and Aspray, W. (1996). *Computer: A History of the Information Machine*. Basic Books, New York.

Cantarello, S., Martini, A., and Nosella, A. (2012). A Multi-Level Model for Organizational Ambidexterity in the Search Phase of the Innovation Process. *Creativity and Innovation Management*, 21(1):28–48.

Ceruzzi, P. E. (2003). *History of Modern Computing*. MIT Press.

Chesbrough, H. W. (2003a). *Open Innovation: The New Imperative for Creating And Profiting from Technology*. Harvard Business School Press.

Chesbrough, H. W. (2003b). The era of open innovation. *MIT Sloan Management Review*, Spring:35–42.

Chesbrough, H. W. (2006). Open innovation: A new paradigm for understanding industrial innovation. In Chesbrough, H. W., Vanhaverbeke, W., and West, J., editors, *Open Innovation Researching a New Paradigm*, chapter 1. Oxford University Press.

Chesbrough, H. W. and Crowther, A. K. (2006). Beyond high tech: early adopters of open innovation in other industries. *R&D Management*, 36(3):229–236.

Chesbrough, H. W., Vanhaverbeke, W., and West, J. (2006). *Open Innovation: Researching a New Paradigm*. Oxford University Press.

Chung, S. A., Singh, H., and Lee, K. (2000). Complementarity, status similarity and social capital as drivers of alliance formation. *Strategic Management Journal*, 22(February 1996):1–22.

Cloodt, M., Hagedoorn, J., and Roijakkers, N. (2010). Inter-firm R&D networks in the global software industry: An overview of major trends and patterns. *Business History*, 52(1):120–149.

Cohen, M. D. and Bacdayan, P. (1994). Organizational routines are stored as procedural memory: Evidence from a laboratory study. *Organization Science*, 5(4):554–568.

Cohen, M. D., Burkhart, R., Dosi, G., Egidi, M., Marengo, L., Warglien, M., and Winter, S. G. (1996). Routines and Other Recurring Action Patterns of Organizations : Contemporary Research Issues. *Industrial and Corporate Change*, 5(3):653–698.

Cohendet, P. and Pénin, J. (2011). Patents to Exclude vs. Include: Rethinking the Management of Intellectual Property Rights in a Knowledge-Based Economy. *Technology Innovation Management Review*, (December):12–17.

Conway, M. E. (1968). How do committees invent. *Datamation*, 14(4):28–31.

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

Cowan, R., Jonard, N., and Zimmermann, J.-B. (2007). Bilateral Collaboration and the Emergence of Innovation Networks. *Management Science*, 53(7):1051–1067.

Dahlander, L. and Gann, D. M. (2010). How open is innovation? *Research Policy*, 39(6):699–709.

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

Dahlin, K. B. and Behrens, D. M. (2005). When is an invention really radical? Defining and measuring technological radicalness. *Research Policy*, 34(5):717–737.

Dalle, J.-M., David, P. A., Besten, M. D., and Steinmueller, W. E. (2008). Empirical issues in open source software. *Information Economics and Policy*, 20(4):301–304.

Darmon, E., Texier, T. L., and Torre, D. (2007). Commercial or open source software? winner-takes-all competition, partial adoption and efficiency. *GREDEG working paper*.

Darmon, E. and Torre, D. (2010). Open source, dual licensing and software competition. *LIXème Congrès annuel de l'AFSE (Association Française de Science Economique)*, pages 1–16.

David, P. A. and Shapiro, J. S. (2008). Community-based production of open-source software: What do we know about the developers who participate? *Information Economics and Policy*, 20(4):364–398.

de Laat, P. B. (2005). Copyright or copyleft? An analysis of property regimes for software development. *Research Policy*, 34(10):1511–1532.

De Nooy, W., Mrvar, A., and Batagelj, V. (2005). *Exploratory network analysis with Pajek*. Cambridge University Press.

de Solla Price, D. J. (1965). Networks of Scientific Papers. *Science*, 149(3683):510–515.

Dodgson, M., Gann, D. M., and Salter, A. (2006). The role of technology in the shift towards open innovation: the case of Procter & Gamble. *R&D Management*, 36(3):333–346.

Dosi, G. (1982). Technological paradigms and technological trajectories: A suggested interpretation of the determinants and directions of technical change. *Research Policy*, 11(3):147–162.

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

Dougherty, D. (1992). Interpretive barriers to successful product innovation in large firms. *Organization Science*, 3(2):179–203.

Eisenhardt, K. M. and Schoonhoven, C. B. (1996). Resource-based view of strategic alliance formation: Strategic and social effects in entrepreneurial firms. *Organization Science*, 7(2):136–151.

Enkel, E., Gassmann, O., and Chesbrough, H. W. (2009). Open R&D and open innovation: exploring the phenomenon. *R&D Management*, 39(4):311–316.

Epicoco, M. (2013). Knowledge patterns and sources of leadership: Mapping the semiconductor miniaturization trajectory. *Research Policy*, 42(1):180–195.

Ernst, H. (2001). Patent applications and subsequent changes of performance: evidence from time-series cross-section analyses on the firm level. *Research Policy*, 30(1):143–157.

Ernst, H. (2003). Patent information for strategic technology management. *World Patent Information*, 25(3):233–242.

Fagerberg, J. (2003). Schumpeter and the revival of evolutionary economics: an appraisal of the literature. *Journal of Evolutionary Economics*, 13(2):125–159.

Fagiolo, G., Moneta, A., and Windrum, P. (2007). A Critical Guide to Empirical Validation of Agent-Based Models in Economics: Methodologies, Procedures, and Open Problems. *Computational Economics*, 30(3):195–226.

Fall, C., Torcsvari, A., Benzineb, K., and Karetka, G. (2003). Automated categorization in the international patent classification. *ACM SIGIR Forum*.

Fleming, L. and Sorenson, O. (2001). Technology as a complex adaptive system: evidence from patent data. *Research Policy*, 30(7):1019–1039.

Fleming, L. and Sorenson, O. (2004). Science as a map in technological search. *Strategic Management Journal*, 25(8-9):909–928.

Fontana, R., Nuvolari, A., and Verspagen, B. (2009). Mapping technological trajectories as patent citation networks. An application to data communication standards. *Economics of Innovation and New Technology*, 18(4):311–336.

Fortuna, M. A., Bonachela, J. A., and Levin, S. A. (2011). Evolution of a modular software network. *Proceedings of the National Academy of Sciences of the United States of America*, 108(50):19985–9.

Fosfuri, A., Giarratana, M. S., and Luzzi, a. (2008). The Penguin Has Entered the Building: The Commercialization of Open Source Software Products. *Organization Science*, 19(2):292–305.

Freeman, C. (2007). The ICT paradigm. In Mansell, R., Avgerou, C., Quah, D., and Silverstone, R., editors, *The Oxford Handbook of Information and Communication Technologies*, chapter 1, pages 34–54. Oxford University Press, Oxford.

Freeman, C. and Louça, F. (2002). *As time goes by: From industrial revolutions to the Information revolutions*. Oxford University Press.

Freeman, C. and Perez, C. (1988). Structural crises of adjustment, business cycles and investment behaviour. In Dosi, G., Freeman, C., Nelson, R. R., and Soete, L., editors, *Technical change and economic theory*. Pinter Publishers, London.

Gale, D. and Shapley, L. (1962). College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15.

Galvin, P. and Morkel, A. (2001). The effect of product modularity on industry structure: the case of the world bicycle industry. *Industry and Innovation*, 8(1):37–41.

Garcia-Vega, M. (2006). Does technological diversification promote innovation? *Research Policy*, 35(2):230–246.

Garrette, B., Castaner, X., and Dussauge, P. (2009). Horizontal alliances as an alternative to autonomous production: Product expansion mode choice in the worldwide aircraft industry 1945-2000. *Strategic Management Journal*, 30(8):885–894.

Garud, R. and Kumaraswamy, A. (1993). Changing competitive dynamics in network industries: An exploration of sun microsystems' open systems strategy. *Strategic Management Journal*, 14(5):351–369.

Garud, R. and Kumaraswamy, A. (1995). Technological and organizational designs for realizing economies of substitution. *Strategic Management Journal*, 16(S1):93–109.

Ghosh, R. A. (2005). An economic basis for open standards, FLOSSPOLS project deliverable D. Technical Report December, MERIT, University of Maastricht.

Gilbert, N. and Terna, P. (2000). How to build and use agent-based models in social science. *Mind & Society*, 1(1):57–72.

Gilsing, V., Nooteboom, B., Vanhaverbeke, W., Duysters, G., and Vandenoord, A. (2008). Network embeddedness and the exploration of novel technologies: Technological distance, betweenness centrality and density. *Research Policy*, 37(10):1717–1731.

Gittelman, M. and Kogut, B. (2003). Does good science lead to valuable knowledge? Biotechnology firms and the evolutionary logic of citation patterns. *Management Science*, 49(4):366–382.

Glott, R. and Ghosh, R. A. (2005). Free / Libre and Open Source Software: Policy Support FLOSSPOLS Skills Survey Interim Report. Technical report, MERIT.

Gomes-Casseres, B., Hagedoorn, J., and Jaffe, A. B. (2006). Do alliances promote knowledge flows? *Journal of Financial Economics*, 80(1):5–33.

Goransson, B. and Soderberg, J. (2005). Long waves and information technologies–on the transition towards the information society. *Technovation*, 25(3):203–211.

Graham, P. (2004). *Hackers & Painters: Big Ideas from the Computer Age*. O'Reilly Media.

Griliches, Z. (1990). Patent statistics as economic indicators: a survey. *Journal of Economic Literature*, 28(4):1661–1707.

Groen, A. J. and Linton, J. D. (2010). Is open innovation a field of study or a communication barrier to theory development? *Technovation*, 30(11):554.

Gulati, R. (1995). Social structure and alliance formation patterns: A longitudinal analysis. *Administrative Science Quarterly*, 40(4):619–652.

Gulati, R., Nohria, N., and Zaheer, A. (2000). Strategic networks. *Strategic Management Journal*, 21(3):203–215.

Gulati, R. and Sytch, M. (2007). Dependence asymmetry and joint dependence in interorganizational relationships: Effects of embeddedness on a manufacturer's performance in procurement. *Administrative Science Quarterly*, 52(1):32–69.

Hagedoorn, J. (1993). Understanding the rationale of strategic technology partnering: Interorganizational modes of cooperation and sectoral differences. *Strategic Management Journal*, 14(5):371–385.

Hagedoorn, J. and Cloodt, M. (2003). Measuring innovative performance: is there an advantage in using multiple indicators? *Research Policy*, 32(8):1365–1379.

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

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

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

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

Hansen, M. T. (2002). The Open Source approach — opportunities and limitations with respect to security and privacy. *Computers & Security*, 21(5):461–471.

Hargadon, A. (2003). *How Breakthrough Happens*. Harvard University Press.

Harhoff, D., M.Scherer, F., and Vopel, K. (2003). Citations, family size, opposition and the value of patent rights. *Research Policy*, 32(8):1343–1363.

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

Hausman, J. (1978). Specification tests in econometrics. *Econometrica*, 46(6):1251–1271.

Hecker, F. (1999). Setting Up Shop: The Business of Open-Source Software. *IEEE Software*, 16(1):45–51.

Helbing, D. and Balietti, S. (2011). How to do agent-based simulations in the future: From modeling social mechanisms to emergent phenomena and interactive systems design. *SFI Working Paper*.

Henderson, R. M. and Clark, K. B. (1990). Architecturai Innovation : The Reconfiguration of Existing Product Technologies and the Failure of Established Firms. *Administrative Science Quarterly*, 35(1):9–30.

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

Henkel, J. and Baldwin, C. Y. (2009). Modularity for Value Appropriation : Drawing the Boundaries of Intellectual Property. *Harvard Business School Working Paper*, 09-097.

Hertel, G., Niedner, S., and Herrmann, S. (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.

Herzog, P. (2008). *Open and Closed Innovation*. Gabler Edition Wissenschaft: Betriebswirtschaftliche Studien in forschungsintensiven Industrien. Gabler.

Hummon, N. and Doreian, P. (1989). Connectivity in a citation network: The development of DNA theory. *Social Networks*, 11(1):39–63.

Hunt, R. (2001). You can patent that? Are patents on computer programs and business methods good for the New Economy? *Business Review*, Q1:5–15.

Hunt, R. and Bessen, J. (2004). The software patent experiment. *Business Review*, Q3:23.

Jaffe, A. B. and Trajtenberg, M. (2005). *Patents, Citations, and Innovations.* The MIT Press.

Jovanovic, B. and Rousseau, P. L. (2005). General purpose technologies. In Aghion, P. and Durlauf, S. N., editors, *Handbook of Economic Growth, Volume 1B*, volume 1, chapter 18. Elsevier B.V.

Jullien, N. and Zimmermann, J.-B. (2009). Firms' contribution to open-source software and the dominant user's skill. *European Management Review*, 6(2):130–139.

Kaplan, S. and Vakili, K. (2012). Identifying Breakthroughs: Using Topic Modeling to Distinguish the Cognitive from the Economic. *DRUID 2012, CBS, Copenhagen, Denmark.*

Klepper, S. (1997). Industry life cycles. *Industrial and corporate change*, 6(1):145–182.

Kogut, B. and Zander, U. (1992). Knowledge of the firm, combinative capabilities, and the replication of technology. *Organization Science*, 3(3):383–397.

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

Kostoff, R. N. (2008). Literature-Related Discovery (LRD): Introduction and background. *Technological Forecasting and Social Change*, 75(2):165–185.

Kostoff, R. N. and Schaller, R. R. (2001). Science and technology roadmaps. *IEEE Transactions on Engineering Management*, 48(2):132–143.

Lakhani, K. and Lonstein, E. (2011). InnoCentive. Com (B). *Harvard Business School General Management Unit Case*, 612-026.

Lane, P. and Lubatkin, M. (1998). Relative absorptive capacity and interorganizational learning. *Strategic management journal*, 19(November 1996):461–477.

Langlois, R. N. and Robertson (1992). Networks and innovation in a modular system: Lessons from the microcomputer and stereo component industries. *Research Policy*, 21(4):297–313.

Lanjouw, J., Pakes, A., and Putnam, J. (1996). How to count patents and value intellectual property: The uses of patent renewal and application data. *NBER Working Paper Series*, 5741.

Lanjouw, J. O. and Schankerman, M. (2004). Patent Quality and Research Productivity: Measuring Innovation with Multiple Indicators. *The Economic Journal*, 114(495):441–465.

Laurent, A. M. S. (2008). *Understanding Open Source and Free Software Licensing*. O'Reilly Media.

Lavie, D. (2007). Alliance portfolios and firm performance: A study of value creation and appropriation in the US software industry. *Strategic Management Journal*, 28(12):1187–1212.

Lavie, D. and Rosenkopf, L. (2006). Balancing exploration and exploitation in alliance formation. *Academy of Management Journal*, 49(4):797–818.

Leiponen, A. E. (2008). Competing Through Cooperation: The Organization of Standard Setting in Wireless Telecommunications. *Management Science*, 54(11):1904–1919.

Lerner, J. (1994). The importance of patent scope: an empirical analysis. *The RAND Journal of Economics*, 25(2):319–333.

Lerner, J. and Tirole, J. (2002). The Simple Economics of Open Source. *The Journal of Industrial Economics*, L(2):197–234.

Levin, R. C., Klevorick, A. K., Nelson, R. R., and Winter, S. G. (1987). Appropriating the returns from industrial research and development. *Brookings papers on Economic Activity*, 3:783.

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

Li, Y., Vanhaverbeke, W., and Schoenmakers, W. (2008). Exploration and Exploitation in Innovation: Reframing the Interpretation. *Creativity and Innovation Management*, 17(2):107–126.

Lippoldt, D. and Stryszowski, P. (2009). *Innovation in the Software Sector*. OECD innovation strategy. OECD Publishing.

MacCormack, A., Rusnak, J., and Baldwin, C. Y. (2006). Exploring the Structure of Complex Software Designs: An Empirical Study of Open Source and Proprietary Code. *Management Science*, 52(7):1015–1030.

Machlup, F. and Penrose, E. (1950). The patent controversy in the nineteenth century. *The Journal of Economic History*, 10(1):1–29.

Malerba, F. (1996). Schumpeterian patterns of innovation are technology-specific. *Research Policy*, 25(3):451–478.

Malerba, F., Nelson, R. R., Orsenigo, L., and Winter, S. G. (2008). Public policies and changing boundaries of firms in a "history-friendly" model of the co-evolution of the computer and semiconductor industries. *Journal of Economic Behavior & Organization*, 67(2):355–380.

Mansfield, E. (1986). Patents and innovation: an empirical study. *Management Science*, 32(2):173–181.

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

March, J. G. (1991). Exploration and exploitation in organizational learning. *Organization science*, 2(1):71–87.

Martinelli, A. (2011). An emerging paradigm or just another trajectory? Understanding the nature of technological changes using engineering heuristics in the telecommunications switching industry. *Research Policy*, 41(2):414–429.

Mazzoleni, R. and Nelson, R. R. (1998a). Economic Theories about the Benefits and Costs of Patents. *Journal of Economic Issues*, XXXII(4):1031–1052.

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

McPherson, A., Proffitt, B., and Hale-Evans, R. (2008). Estimating the Total Development Cost of a Linux Distribution. Technical Report October, The Linux Foundation.

Mina, A., Ramlogan, R., Tampubolon, G., and Metcalfe, J. (2007). Mapping evolutionary trajectories: Applications to the growth and transformation of medical knowledge. *Research Policy*, 36(5):789–806.

Mori, S., Suen, C. Y., and Yamamoto, K. (1992). Historical review of OCR research and development. *Proceedings of the IEEE*, 80(7).

Moser, P. (2003). How do patent laws influence innovation? Evidence from nineteenth-century world fairs. *NBER Working Paper Series*, 9909:1–54.

Moser, P. (2007). Why don't inventors patent? *NBER Working Paper Series*, 13294:1–50.

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

Mowery, D. C., Oxley, J. E., and Silverman, B. S. (1998). Technological overlap and interfirm cooperation: implications for the resource-based view of the firm. *Research Policy*, 27(5):507–523.

Nash, J. C. (2010). Directions for open source software over the next decade. *Futures*, 42(4):427–433.

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

Nelson, R. R. and Winter, S. G. (1982). *An Evolutionary Theory of Economic Change*. Belknap Press of Harvard University Press.

Nerkar, A. (2003). Old is gold? The value of temporal exploration in the creation of new knowledge. *Management Science*, 49(2):211–229.

Nesta, L. and Mangematin, V. (2002). Industry Life Cycle, Knowledge Generation and Technological Networks. In *Networks, Alliances and Partnerships in the Innovation Process*. Kluwer Academic, Dordrecht.

Nohria, N. and Garcia-Pont, C. (1991). Global strategic linkages and industry structure. *Strategic Management Journal*, 12(S1):105–124.

Nonaka, I. (1994). A dynamic theory of organizational knowledge creation. *Organization science*, 5(1):14–37.

Nooteboom, B., Vanhaverbeke, W., Duysters, G., Gilsing, V., and Vandenoord, A. (2007). Optimal cognitive distance and absorptive capacity. *Research Policy*, 36(7):1016–1034.

Ozman, M. (2005). *Networks, organizations and knowledge*. PhD thesis, Maastricht Economics Research Instituet on Innovation and Technology.

Ozman, M. (2010). The Joint Effect of Technological Distance and Market Distance on Strategic Alliances. *Working Papers of BETA*, 22.

Ozman, M. (2011). Modularity, industry life cycle and open innovation. *Journal of Technology Management & Innovation*, 6(1):26–37.

Ozman, M. and Özaygen, A. (2011). The Role of Technological and Market Complementarities in Strategic Alliances. *EMEA 2011 "Evolutionary perspectives on technological change and industrial dynamics", LEM-Scuola Superiore Sant'Anna, Pisa, 14-16 February 2011.*

O'Shaughnessy, D. (2008). Invited paper: Automatic speech recognition: History, methods and challenges. *Pattern Recognition*, 41(10):2965–2979.

Page, S. E. (2005). Agent-based models. In *The New Palgrave Dictionary of Economics*, pages 1–12. Palgrave MacMillan.

Panzar, J. and Willig, R. (1981). Economies of scope. *The American Economic Review*, 71(2):268–272.

Parnas, D. L. (1972). On the criteria to be used in decomposing systems into modules. *Communications of the ACM*, 15(12):1053–1058.

Parnas, D. L., Clements, P. C., and Weiss, D. M. (1985). The Modular Structure of Complex Systems. *IEEE Transactions on Software Engineering*, 11(3):259–266.

Pentland, B. T. and Feldman, M. S. (2005). Organizational routines as a unit of analysis. *Industrial and Corporate Change*, 14(5):793–815.

Perez, C. (1983). Structural change and assimilation of new technologies in the economic and social systems. *Futures*, 15(5):357–375.

Perez, C. (2002). *Technological Revolutions and Financial Capital*. Edward Elgar, Chaltenham, UK.

Poppo, L. and Zenger, T. (1998). Testing alternative theories of the firm: transaction cost, knowledge based, and measurement explanations for make or buy decisions in information services. *Strategic Management Journal*, 19(9):853–877.

Pyka, A. (2000). Informal networking and industrial life cycles. *Technovation*, 20(1):25–35.

Quintana-Garcia, C. and Benavides-Velasco, C. A. (2008). Innovative competence, exploration and exploitation: The influence of technological diversification. *Research Policy*, 37(3):492–507.

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

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

Rosenkopf, L. and Nerkar, A. (2001). Beyond local search: boundary-spanning, exploration, and impact in the optical disk industry. *Strategic Management Journal*, 22(4):287–306.

Rothaermel, F. T. and Boeker, W. (2008). Old technology meets new technology: complementarities, similarities, and alliance formation. *Strategic Management Journal*, 29(1):47–77.

Rowley, T. J., Behrens, D. M., and Krackhardt, D. (2000). Redundant governance structures: an analysis of structural and relational embeddedness in the steel and semiconductor industries. *Strategic Management Journal*, 21(3):369–386.

Sanchez, R. and Mahoney, J. T. (1996). Modularity, flexibility and knowledge management in product and organization design. *Strategic Management Journal*, 17(WINTER):63–76.

Sanderson, S. and Uzumeri, M. (1995). Managing product families: The case of the Sony Walkman. *Research Policy*, 24(5):761–782.

Satoh, S., Nakamura, Y., and Kanade, T. (1999). Name-it: Naming and detecting faces in news videos. *Multimedia, IEEE*, 6(1):22–35.

Saviotti, P. P. and Metcalfe, J. (1984). A theoretical approach to the construction of technological output indicators. *Research Policy*, 13(3):141–151.

Schelling, T. M. (1978). *Micromotive and Macrobehavior*. W. W. Norton & Company, Inc., New York.

Schilling, M. A. (2000). Toward a General Modular Systems Theory and Its Application to Interfirm Product Modularity. *The Academy of Management Review*, 25(2):312.

Schmoch, U. (2008). Concept of a Technology Classification for Country Comparisons, Final Report to the World Intellectual Property Organisation (WIPO). Technical Report June, Fraunhofer Institute for Systems and Innovation Research, Karlsruhe, Germany.

Schoenmakers, W. and Duysters, G. (2006). Learning in strategic technology alliances. *Technology Analysis & Strategic Management*, 18(2):245 –264.

Schroll, A. and Mild, A. (2012). A critical review of empirical research on open innovation adoption. *Journal für Betriebswirtschaft*, 62(2):85–118.

Shan, W., Walker, G., and Kogut, B. (1994). Interfirm cooperation and startup innovation in the biotechnology industry. *Strategic Management Journal*, 15(5):387–394.

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

Shapiro, C. and Varian, H. R. (1999). The art of standards wars. *California Management Review*, 41(2):8–32.

Shibata, N., Kajikawa, Y., Takeda, Y., and Matsushima, K. (2008). Detecting emerging research fronts based on topological measures in citation networks of scientific publications. *Technovation*, 28(11):758–775.

Simcoe, T. (2006). Open standards and intellectual property rights. In Chesbrough, H. W., Vanhaverbeke, W., and West, J., editors, *Open Innovation Researching a New Paradigm*. Oxford University Press.

Simon, H. A. (1962). The architecture of complexity. *Proceedings of the American Philosophical Society*, 106(6):467–482.

Snoek, C. and Worring, M. (2005). Multimodal video indexing: A review of the state-of-the-art. *Multimedia Tools and Applications*, 25(1):5–35.

Spender, J. and Grant, R. M. (1996). Knowledge and the firm: overview. *Strategic Management Journal*, 17:109–122.

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

Steinmueller, W. E. (1995). The US software industry: an analysis and interpretive history. In Mowery, D. C., editor, *The International Computer Software Industry*, pages 15–40. Oxford University Press.

Steinmueller, W. E. (2007). The economics of ICTs: Building Blocks and Implications. In Mansell, R., Avgerou, C., Quah, D., and Silverstone, R., editors, *The Oxford Handbook of Information and Communication Technologies*, chapter 8, pages 196–224. Oxford University Press.

Stevnsborg, N. and de La Potterie, B. V. P. (2007). *Patenting Procedures and Filing Strategies at the EPO*. Oxford University Press.

Su, H.-N. and Lee, P.-C. (2012). Framing the structure of global open innovation research. *Journal of Informetrics*, 6(2):202–216.

Suárez, F. F. (2004). Battles for technological dominance: an integrative framework. *Research Policy*, 33(2):271–286.

Suárez, F. F. and Utterback, J. M. (1995). Dominant designs and the survival of firms. *Strategic Management Journal*, 16(6):415–430.

Teece, D. J. (1980). Economies of scope and the scope of the enterprise. *Journal of Economic Behavior & Organization*, 1(3):223–247.

Teece, D. J. (1986). Profiting from technological innovation: Implications for integration, collaboration, licensing and public policy. *Research Policy*, 15(6):285–305.

Trajtenberg, M. (1990). A Penny for Your Quotes: Patent Citations and the Value of Innovations. *The RAND Journal of Economics*, 21(1):172–187.

Tseng, Y., Lin, C., and Lin, Y. (2007). Text mining techniques for patent analysis. *Information Processing & Management*, 43(5):1216–1247.

Tushman, M. L. and O'Reilly III, C. A. (1996). Ambidextrous organizations: Managing evolutionary and revolutionary change. *California Management Review*, 38(4):8–30.

Ulrich, K. (1995). The role of product architecture in the manufacturing firm. *Research Policy*, 24(3):419–440.

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

Utterback, J. M. (1996). *Mastering the Dynamics of Innovation.* Harvard Business School Press.

van der Heijden, S. (2010). Landscaping of an Emerging Technology by Bibliometric Analyses; A Case Study on Wireless Energy Transfer. *Master Thesis, Eindhoven University of Technology.*

van Zeebroeck, N. (2011). The puzzle of patent value indicators. *Economics of Innovation and New Technology*, 20(1):33–62.

Vanhaverbeke, W. (2006). The interorganizational context of open innovation. In Chesbrough, H. W., Vanhaverbeke, W., and West, J., editors, *Open Innovation Researching a New Paradigm*, chapter 11, pages 205–219. Oxford University Press.

Varian, H. R. and Shapiro, C. (2003). Linux Adoption in the Public Sector: An Economic Analysis. *Working Paper, University of California Berkeley*, pages 1–26.

Verspagen, B. (2007). Mapping technological trajectories as patent citation networks: A study on the history of fuel cell research. *Advances in Complex Systems*, 10(1):93–115.

von Hippel, E. (2010). Comment on "Is open innovation a field of study or a communication barrier to theory development?". *Technovation*, 30(11):555.

Wang, Y., Liu, Z., and Huang, J. (2000). Multimedia content analysis-using both audio and visual clues. *IEEE Signal Processing Magazine*, 17(6):12–36.

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

West, J. (2003). How open is open enough? Melding proprietary and open source platform strategies. *Research Policy*, 32(7):1259–1285.

West, J. and Gallagher, S. (2006). Patterns of open innovation in open source software. In Chesbrough, H. W., Vanhaverbeke, W., and West, J., editors, *Open Innovation Researching a New Paradigm*, chapter 5. Oxford University Press.

Wiersema, M. and Bowen, H. (2008). Corporate diversification: the impact of foreign competition, industry globalization, and product diversification. *Strategic Management Journal*, 132(August 2007):115–132.

Windrum, P., Diaz, C., and Filiou, D. (2009). Exploring the relationship between technical and service characteristics. *Journal of Evolutionary Economics*, 19(4):567–588.

Yin, R. K. (2003). *Case Study Research: Design and Methods*. Applied Social Research Methods. SAGE Publications, 3rd edition.

Appendices

Appendix A

Additional outputs, Chapter 3

Table A.1: Summary statistics for $\alpha = 0.1$ and $\beta = 2$.

Type	Firm #	$\sum_i^N \phi_i$	$\sum_i^N \phi_i$ / Firm #	Alliance #	Alliance # / Firm
I	34.60	5602.70	161.90	9944.80	287.40
II	35.40	3430.60	96.90	5280.40	149.20
III	30.00	2886.10	96.20	4194.80	139.80

Table A.2: Summary statistics for $\alpha = 0.1$ and $\beta = 4$.

Type	Firm #	$\sum_i^N \phi_i$	$\sum_i^N \phi_i$ / Firm #	Alliance #	Alliance # / Firm
I	37.40	29832.30	797.70	13742.20	367.40
II	29.20	12790.50	438.00	5657.60	193.80
III	33.40	13043.60	390.50	4933.40	147.70

Table A.3: Summary statistics for $\alpha = 0.1$ and $\beta = 6$.

Type	Firm #	$\sum_i^N \phi_i$	$\sum_i^N \phi_i$ / Firm #	Alliance #	Alliance # / Firm
I	37.00	28181.30	761.70	13368.00	361.30
II	33.00	18599.90	563.60	5396.00	163.50
III	30.00	17107.00	570.20	3080.00	102.70

Table A.4: Summary statistics for $\alpha = 0.2$ and $\beta = 2$.

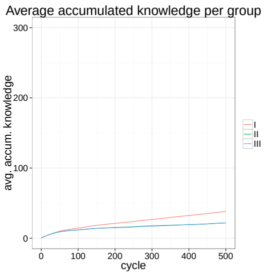
Type	Firm #	$\sum_i^N \phi_i$	$\sum_i^N \phi_i$ / Firm #	Alliance #	Alliance # / Firm
I	34.60	4521.10	130.70	9050.60	261.60
II	33.60	1828.00	54.40	3258.40	97.00
III	31.80	1407.60	44.30	2034.20	64.00

Table A.5: Summary statistics for $\alpha = 0.2$ and $\beta = 4$.

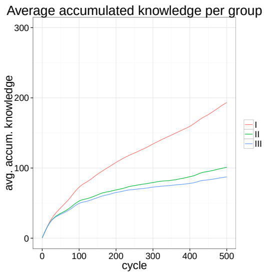
Type	Firm #	$\sum_i^N \phi_i$	$\sum_i^N \phi_i$ / Firm #	Alliance #	Alliance # / Firm
I	35.80	18080.00	505.00	11720.60	327.40
II	32.80	6353.50	193.70	3664.20	111.70
III	31.40	5716.70	182.10	2369.60	75.50

Table A.6: Summary statistics for $\alpha = 0.2$ and $\beta = 6$.

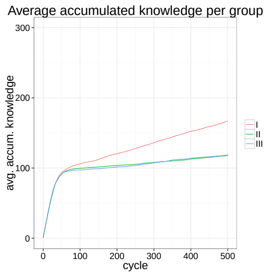
Type	Firm #	$\sum_i^N \phi_i$	$\sum_i^N \phi_i$ / Firm #	Alliance #	Alliance # / Firm
I	38.00	16265.30	428.00	12178.40	320.50
II	31.40	10336.60	329.20	3722.20	118.50
III	30.60	8970.90	293.20	1908.20	62.40



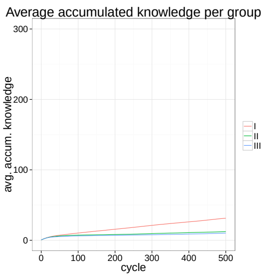
(a) $\alpha = 0.1, \beta = 2$



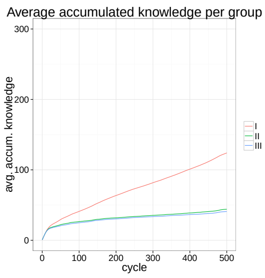
(b) $\alpha = 0.1, \beta = 4$



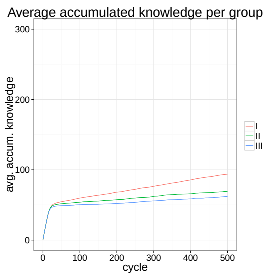
(c) $\alpha = 0.1, \beta = 6$



(d) $\alpha = 0.2, \beta = 2$

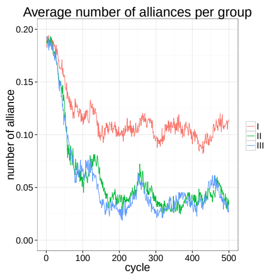


(e) $\alpha = 0.2, \beta = 4$

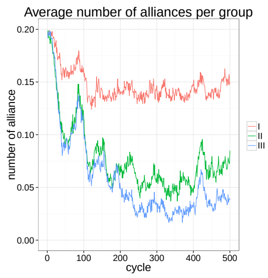


(f) $\alpha = 0.2, \beta = 6$

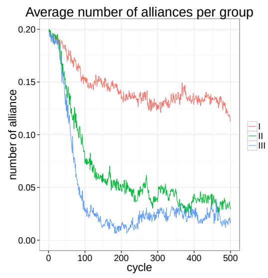
Figure A.1: Average accumulated knowledge.



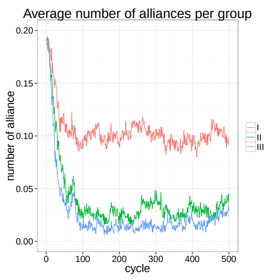
(a) $\alpha = 0.1, \beta = 2$



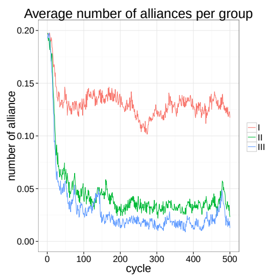
(b) $\alpha = 0.1, \beta = 4$



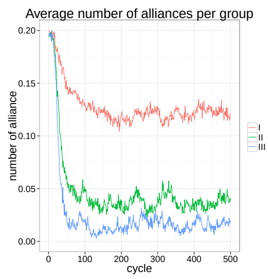
(c) $\alpha = 0.1, \beta = 6$



(d) $\alpha = 0.2, \beta = 2$

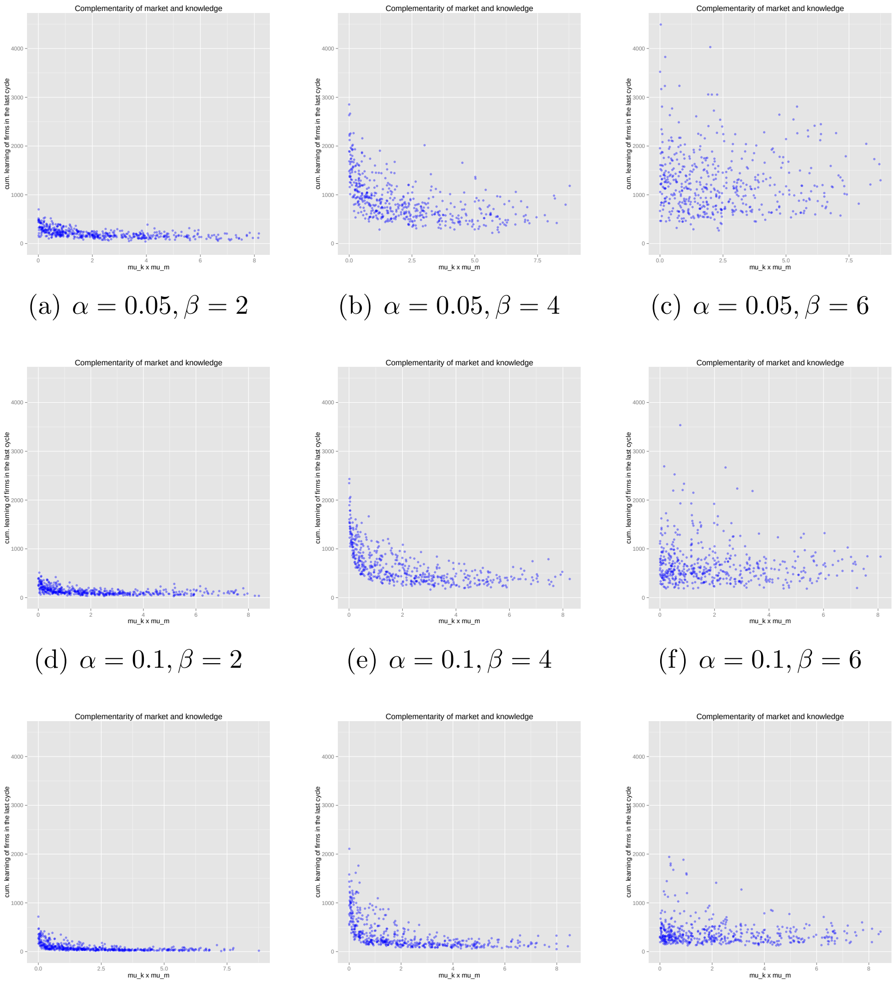


(e) $\alpha = 0.2, \beta = 4$



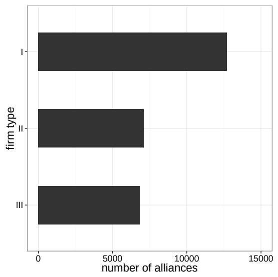
(f) $\alpha = 0.2, \beta = 6$

Figure A.2: Average alliances.

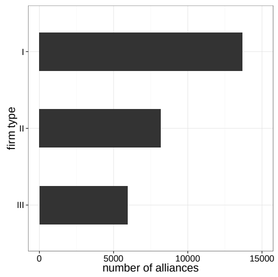


(a) $\alpha = 0.05, \beta = 2$
(b) $\alpha = 0.05, \beta = 4$
(c) $\alpha = 0.05, \beta = 6$
(d) $\alpha = 0.1, \beta = 2$
(e) $\alpha = 0.1, \beta = 4$
(f) $\alpha = 0.1, \beta = 6$
(g) $\alpha = 0.2, \beta = 2$
(h) $\alpha = 0.2, \beta = 4$
(i) $\alpha = 0.2, \beta = 6$

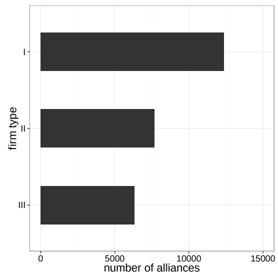
Figure A.3: Effect of $\sigma_m \dot{\sigma}_k$ on learning.



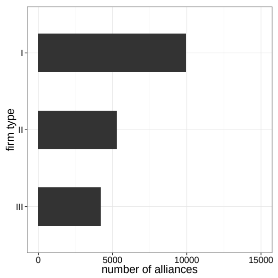
(a) $\alpha = 0.05, \beta = 2$



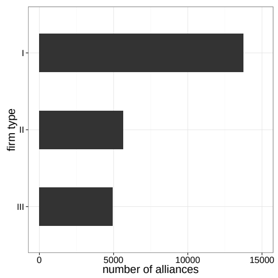
(b) $\alpha = 0.05, \beta = 4$



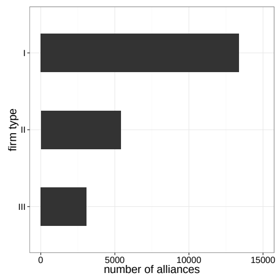
(c) $\alpha = 0.05, \beta = 6$



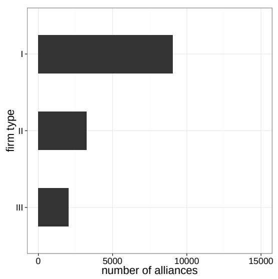
(d) $\alpha = 0.1, \beta = 2$



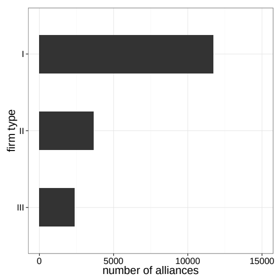
(e) $\alpha = 0.1, \beta = 4$



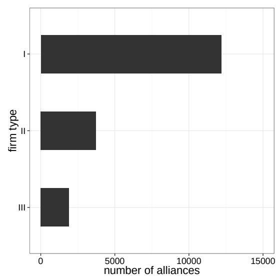
(f) $\alpha = 0.1, \beta = 6$



(g) $\alpha = 0.2, \beta = 2$

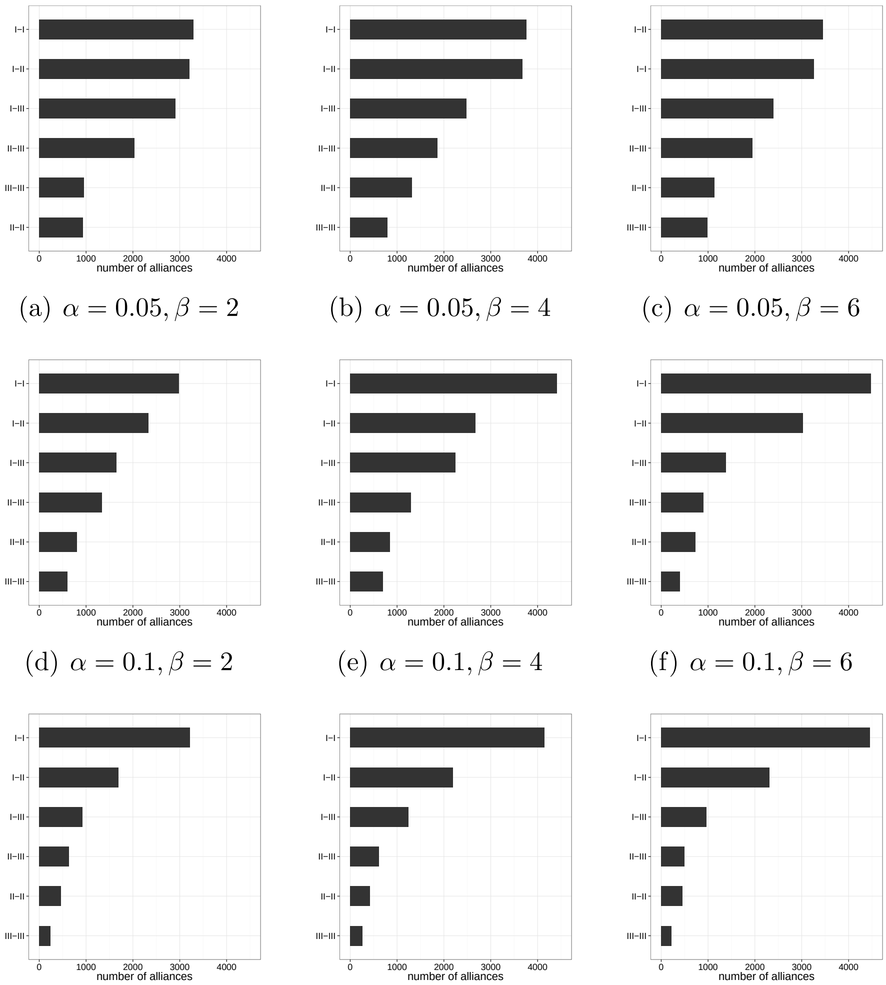


(h) $\alpha = 0.2, \beta = 4$



(i) $\alpha = 0.2, \beta = 6$

Figure A.4: Number of alliances.



(a) $\alpha = 0.05, \beta = 2$ (b) $\alpha = 0.05, \beta = 4$ (c) $\alpha = 0.05, \beta = 6$

(d) $\alpha = 0.1, \beta = 2$ (e) $\alpha = 0.1, \beta = 4$ (f) $\alpha = 0.1, \beta = 6$

(g) $\alpha = 0.2, \beta = 2$ (h) $\alpha = 0.2, \beta = 4$ (i) $\alpha = 0.2, \beta = 6$

Figure A.5: Number of alliances between different type of firms.

Appendix B

Lists of most cited patents, Chapter 5

Table B.1: Top 20 cited patents related to character recognition patent group and their assignees.

# of citations	patent num	Pub. year	Assignees name
595	5692073	1997	Xerox
586	4916441	1990	CliniCom Inc.
580	5051736	1991	IBM
489	5677955	1997	Bell Communications Research,Financial Services Technology Consortium,The First National Bank of Boston
390	4387297	1983	Symbol Technologies
312	4409470	1983	Symbol Technologies
309	5721788	1998	Corbis
299	5410141	1995	Norand
297	5978773	1999	NeoMedia Technologies
281	5378883	1995	Omniplanar Inc.
276	6964374	2005	Lucent Technologies Inc.
268	5354977	1994	Roustaei; Alex
257	5804803	1998	IBM
235	6400996	2002	Hoffberg, Hoffberg-Borghesani
234	5756981	1998	Symbol Technologies
230	4264808	1981	NCR
216	5532467	1996	Roustaei; Alex
211	5337361	1994	Symbol Technologies
210	4972496	1990	Grid Systems
206	6311214	2001	Digimarc

Table B.2: Top 20 cited patents related to audio and speech analysis patent group and their assignees.

# of citations	patent num	Pub. year	Assignees name
797	5892900	1999	InterTrust Technologies
591	5247347	1993	Bell Atlantic Network Services
504	5086385	1992	Custom Command Systems
503	5675507	1997	Bobo, II; Charles R.
472	5732074	1998	CellPort Labs
461	5353331	1994	Bell Atlantic Network Services
460	5335276	1994	Texas Instruments
446	4757267	1988	Applied Telematics
424	5327486	1994	Bell Communications Research
405	4949187	1990	Cohen; Jason M.
404	5721827	1998	Logan; James
392	5915001	1999	Vois
382	4305131	1981	Best; Robert M.
365	5652789	1997	Wildfire Communications
356	5884262	1999	Bell Atlantic Network Services
356	5410343	1995	Bell Atlantic Network Services
350	5297031	1994	Chicago Board of Trade
337	5334974	1994	Moore, Jr.; Daniel D.,Simms; Charles G.,Simms; James R.
332	5497373	1996	Ericsson Messaging Systems Inc.
322	5132992	1992	Browne; H. Lee,Yurt; Paul

Table B.3: Top 20 cited patents related to image analysis patent group and their assignees.

# of citations	patent num	Pub. year	Assignees name
620	4870302	1989	Xilinx
333	5613012	1997	Smarttouch
289	6161130	2000	Microsoft
288	5963134	1999	Checkpoint Systems
280	5410344	1995	Arrowsmith Technologies
272	5765176	1998	Xerox
265	5862260	1999	Digimarc
261	4582985	1986	Lofberg, Bo
248	6236365	2001	TracBeam
242	6122403	2000	Digimarc
239	5446891	1995	IBM
235	6400996	2002	Hoffberg ,Hoffberg-Borghesani
226	5440723	1995	IBM
226	5185667	1993	TeleRobotics International
225	5892903	1999	Internet Security Systems
219	4965725	1990	Nueromedical Systems
219	5313953	1994	InControl
206	5893095	1999	Virage
206	4553261	1985	Froessl, Horst
204	6405132	2002	Intelligent Technologies International

Table B.4: Top 20 cited patents related to video indexing patent group and their assignees.

# of citations	patent num	Pub. year	Assignees name
205	5014267	1991	Datapoint
185	5253275	1993	Browne, H. Lee
148	5666157	1997	ARC
145	5969755	1999	Texas Instruments
129	6236395	2001	Sharp Labs. of America
126	5227985	1993	University of Maryland
123	5454043	1995	Mitsubishi Electric Research Labs.
123	5774591	1998	Xerox
122	4930160	1990	Vogel; Peter S.
115	6301370	2001	Eyematic Interfaces
112	5442389	1995	AT&T
108	5835616	1998	University of Central Florida
104	4331974	1982	Iri
97	5012522	1991	The USA Air
97	5781650	1998	University of Central Florida
96	5245533	1993	A. C. Nielsen Comp.
94	5412738	1995	Istituto Trentino Di Cultura
93	5430809	1995	Sony
93	6374260	2002	Magnifi
91	5423554	1995	MetaMedia Ventures

Appendix C

Case study questions, Chapter 6

Bellow are the main questions asked to the members of the software R&D members during the open ended interviews. Various questions are also asked with respect to the answers obtained.

Questions related to the video indexing project.

- How many people are working on this project?
- Are all modules made within Alcatel Bell-Labs?
- Do you expect that main modules will be finished within Alcatel?
- On what part of the project your main university partner is working on?
- Could you re-frame what is expected from your team and your university partner and (if any) other partners to attain at the end of this project?
- Do you have a distinct map of team and the modules that these teams are working on?
- Do you target a distinct market niche with this product?

- In which markets do you expect that this product will be used?

Knowledge inflow and outflow.

- How do you find any business partner?
- Do you develop any business model for your R&D project?
- Do you make any research for your future and possible partners?
- Do you have any SMEs as an R&D partner?
- How do you distinguish any start-up which works in the same field as yours?
- How do you deal with the IPR issues related to your project?
- Which firms do you see as your main competitor?
- Do you make any research about your competitors?

FLOSS and open standards.

- Do you develop a new standard for the video indexing project?
- If open standards are used why do you prefer them?
- Do you use any FLOSS code in your project?
- Do you collaborate with any FLOSS developer group?
- How do you trace the frontier between the close code software and FLOSS code in your project?