Industrial Structure and Policy Choice

Notes on the Evolution of Semiconductors and Open Source

Ilkka Tuomi

DRAFT WORKING PAPER

Joint Research Centre
Institute for Prospective Technological Studies
5 June 2004
ilkka.tuomi@cec.eu.int

Historical analysis of the development of semiconductor industry in Silicon Valley shows that patenting and licensing policies have had a profound impact on the evolution of ICT industry and regional growth. Weak intellectual property rights and social networks with strong ties seem to underlie the growth dynamic in Silicon Valley in the 1950s and 1960s. Independent of institutional and social factors, a key developmental factor has also been the exceptional technological trajectory of semiconductors.

Software technology has many similarities with semiconductor technology. One may therefore ask, what an analysis of the history of the development of semiconductor industry could tell us about the future of software industry. In particular, an interesting question is whether institutional characteristics, including the regime of intellectual property rights, could be organized so that rapid growth would be possible in software industries in the future.

The views expressed in this draft working paper are intended to promote discussion and research. They do not represent the views of the Joint Research Centre, the Institute for Prospective Technological Studies, or the European Commission.

Industrial Structure and Policy Choice: Notes on the Evolution of Semiconductors and Open Source

Abstract

Historical analysis of the development of semiconductor industry in Silicon Valley shows that patenting and licensing policies have had a profound impact on the evolution of ICT industry and regional growth. Weak intellectual property rights and social networks with strong ties seem to underlie the growth dynamic in Silicon Valley in the 1950s and 1960s. Independent of institutional and social factors, a key developmental factor has also been the exceptional technological trajectory of semiconductors.

Software technology has many similarities with semiconductor technology. One may therefore ask, what an analysis of the history of the development of semiconductor industry could tell us about the future of software industry. In particular, an interesting question is whether institutional characteristics, including the regime of intellectual property rights, could be organized so that rapid growth would be possible in software industries in the future.

Introduction

The reasons for the rapid growth and high concentration of semiconductor, computer, and computer networking firms in the Silicon Valley area have attracted much attention during the last two decades. Policy makers, in particular, have been interested in the factors that have made Silicon Valley one of the global focal points in computing technologies.

The early policy-related analyses of Silicon Valley phenomenon emphasized characteristics and factors that seemed to underlie economic growth in Silicon Valley. Analysts came out with recommendations to physically concentrate innovative activities close to universities, to link university research with entrepreneurial activities, to facilitate access to venture funding and to subsidize infrastructures that was needed for research and business creation. In essence, the analysts came up with lists of critical success factors that seemed to underlie the Silicon Valley model. By creating locations where these critical success factors were available, policy makers expected to promote R&D intensive economic growth. These initiatives rarely were successful (Castells & Hall, 1994).

More recently, studies on innovative regions have focused on the importance of underlying institutional and social structures. For example, it has been shown that Silicon Valley is a complex system of interdependent actors, where networks of firms, people, and knowledge are deeply embedded and where the actors are connected with strong ties. Researchers have emphasized institutional linkages between start-up firms, venture capitalists, law firms, investment bankers, and the leading firms and university labs (Saxenian, 1994; Kenney, 2000), and pointed out that social networks have been central for the development of the region (Saxenian, 1999 Castilla, Hwang, Granovetter, & Granovetter, 2000).

An underlying view in this paper is that we need to move beyond static descriptive accounts of the Silicon Valley if we want to understand its development. Discussions on development require dynamic and historical descriptions. Instead of asking what Silicon Valley is we need to ask how it became what it is. This leads to a dynamic and historical socio-economic characterization of the Silicon Valley phenomenon. Conceptually, the outcome of such an analysis is a model that emphasizes generative processes that over time lead to development, and a better understanding of the factors that inhibit and enforce specific types of developmental paths. Instead of asking whether we can replicate Silicon Valley in other geographical locations, we should ask whether the dynamic that has been driving growth in Silicon Valley could be replicated in other settings.

In particular, I will be asking the question whether this growth dynamic could occur in open source software industries. In this context, it will be interesting to analyze whether the Silicon Valley growth dynamic, or—more exactly—the early history of the semiconductor industry could be repeated independent of geographic location.

At one level, this paper therefore revisits the contentious issue of "replicating the Silicon Valley success story" in different geographical and historical contexts. I will argue that attempts to copy the model have failed mainly because they have inadequately analyzed the dynamic of development, assuming that the present configuration of developmental factors is sufficient to explain the phenomenon. I will also argue that although historical accounts of the development of Silicon Valley are numerous, and shed light on important factors on technology development and economic growth, they have not adequately integrated historical facts with theory of innovation. On the contrary, implicit ideas about the process of innovation have often led to reorganization of historical facts about the Silicon Valley story so that they fit conventional conceptions about technology development. Historical accounts of the creation of the Silicon Valley phenomenon have sometimes been retrospective projections where details have become reordered and forgotten so that the underlying story can be made understandable and logical. In this process, the logic, however, sometimes has become described in terms that miss perhaps essential points.

The historical development of Silicon Valley has several qualitatively different phases, and the logic of development has varied during the different decades. As its name reveals, the roots of Silicon Valley are, however, deeply grounded in the semiconductor industry. In this paper I will therefore focus on the dynamics of development of the semiconductor industry and in particular the period of 1960s. Using the early history of semiconductor industry as a model, I will then ask could this historical growth pattern be repeated in software industries. In particular, I try to highlight the potential importance of intellectual property regimes by drawing an analogy between the histories of Silicon Valley and the Linux operating system development.

This paper is an early working draft. It touches many key issues only superficially, and explores only the first steps of a path that needs to be traversed to fully justify some of the claims made. In its present form, the paper tries to suggest some possible futures for software industry, and show that institutional structures such as intellectual policy regimes may have important consequences for the future evolution of information and communication technologies. The current paper, therefore, develops the point advocated by Merges (1996), who argued that intellectual property policies may have important influence in the innovativeness, vitality, and industry structure in software industries. The present paper also tries to shed some light on the linkages between innovation in software systems, location and outsourcing of software activities and growth dynamics in this industry.

The basic argument that underlies the present paper is a simple one. The rapid growth and concentration of semiconductor industry in Silicon Valley has been driven by organic socio-economic growth where spin-offs have played a central role. Spin-offs implement a specific form of innovation dynamic that needs to be distinguished from a combinatorial mode of innovation. The spin-off dynamic strongly depends on institutional and social conditions that can either support of constrain growth. In Silicon Valley, growth has been facilitated, as knowledge and competences have relatively easily leaked out of the parent firms where they have originally been created. This spin-off dynamic has been possible only because the underlying technology has for several decades provided a continuous technology trajectory, and because strong intellectual property right regime was not enforced in Silicon Valley during the first two decades of growth. A similar dynamic underlies the development and rapid growth of the GNU/Linux operating system, where intellectual property rights have intentionally been designed to support growth. From a knowledge and competence creation point of view, the early history of semiconductor technology and the history of Linux therefore resemble each other. An empirical analysis of software industry in Europe shows that the industry growth path does not follow this growth dynamic, but that it could perhaps be changed by policies that are informed by insights of the similarities and differences between the Silicon Valley growth dynamic and the current growth dynamic of software industries.

Revisiting the Silicon Valley story

The main characteristics of the historical development of semiconductor industry in Silicon Valley are well known.[1] For the purposes of the present paper, however, only a brief and condensed summary is needed.

In 1954, William Shockley returned back to California, where he set up the Shockley Semiconductor Laboratory, a division of Beckman Instruments, with a goal to develop silicon-based transistors.[2]

Shockley Semiconductor Laboratory started in the beginning of 1956 and it grew quickly. In autumn 1957 it had about fifty employees. Among the first was Gordon

[1] In fact, the story of Silicon Valley has been told so often that today it is surprisingly difficult to verify the various versions of the history. Although the main characteristics of the story are often repeated, the details are also contradictory and do not always support the narrative.

[2] During 1954 Shockley was visiting lecturer at the California Institute of Technology, and in 1954-55 Deputy Director and Research Director of Weapons System Evaluation Group in the Defense Department. According to different histories, Shockley left Bell Telephone Laboratories in 1953, 1954 or 1955. Shockley Semiconductor Laboratory was set up in Menlo Park, next to Palo Alto where Shockley had grown up and where his mother lived. According to different histories, Shockley Semiconductor Laboratory was founded in 1955 or in February 1956. In 1958, Shockley Semiconductor Laboratory became Shockley Transistor Corporation, a subsidiary of Beckman Instruments. Shockley was originally planning to start to produce transistors but quickly changed his mind and decided to pursue four layer diodes instead.

Moore, a chemist who had been looking for a job at Lawrence Livermore Laboratory, located on the east side of the San Francisco Bay. Through some unknown procedure, Shockley got access to the records of people who had been interviewed for jobs at Lawrence Livermore Labs, called Moore one evening, and offered him a job at Shockley Semiconductors. Moore accepted, and returned back to his roots.[3]

Shockley's difficult management style created numerous conflicts during the first year of the operation. As a result, a group of Shockley's employees tried to convince Beckman that Shockley Laboratories would need a new director.[4] When this did not succeed, they concluded that they had burnt too many bridges at Shockley, and started to look for another firm that could hire them as a group to do work with transistors. Gene Kleiner, one of the group, had a friend who worked at a New York investment bank. A senior partner of the bank and a young Harvard MBA Arthur Rock visited the group in California, and suggested after some discussions that a better idea would be that the group would set up their own firm. As the members of the group already had houses in the Stanford area and had settled down there, they thought the idea was good. (Moore, 1995)

The entrepreneurs went through a listing of New York Stock Exchange in the Wall Street Journal, trying to locate all possible firms that might be interested in funding a new semiconductor operation. They found about thirty-five potential candidates, and gave the list to the investment bank. The bankers visited all the candidates to pitch the proposal. All declined the offer. By accident, the investment bank discussed the idea with Sherman Fairchild, the founder of Fairchild Camera and Instrument Corporation, who got interested. As a result, Fairchild Camera and Instrument Corporation decided to set up a new division, Fairchild Semiconductor Corporation. It started its operations in 1957, with $3,500 in capital.

The eight founders of Fairchild Semiconductor believed that they needed a competent general manager to run the business. After some searching, they hired Ed Baldwin from Hughes Semiconductor. In about a year, Baldwin left the company to set up his own firm, Rheem Semiconductor, taking with him ten Fairchild employees. By 1986 at least 31 semiconductor companies could directly be traced to Fairchild. Some of this spin-offs were hostile, others simply focused on opportunities that Fairchild could not address.[5] As Moore put it:

[3] Moore was born in Pescadero, close to San Francisco, lived as a child in Redwood City, studied as undergraduate at UC Berkeley, and did his Ph.D. at California Institute of Technology. As it was difficult to find technical work in the San Francisco area, he took a post in Applied Physics Laboratory of Johns Hopkins University in Silver Spring, Maryland, just outside of Washington, DC. Moore started there in September of 1953, and returned to the Stanford area about two and half years later. (Moore, 1995).

[4] The group is commonly called the traitorous eight. It consisted of Gordon Moore, Julius Blank, Victor Grinich, Jean Hoerni, Gene Kleiner, Jay Last, Robert Noyce, and Sheldon Roberts.

[5] Fairchild spin-off companies include Rheem, Advanced Micro Devices Inc (AMD), LSI Logic, Teledyne, National Semiconductor., Intersil, and Intel, and also venture capital firms such as Kleiner Perkins and semiconductor industry suppliers such as Electroglass Corporation. Fairchild sued Rheem, claiming that it had stolen Fairchild's "cook-book." The judge was not exceptionally supportive for this claim, as Fairchild has done something similar with Shockley, and the case was settled outside court. Fairchild Semiconductor itself was sold to National Semiconductor in 1987, which absorbed its operations and stopped using its name. Fairchild Semiconductor re-emerged in 1997 through what has been called the first management buy-out in semiconductor industry.

> In any case, integrated circuits, MOS transistors, and the like proved too rich a vein for a company the size of Fairchild to mine, resulting in what came to be termed the ‘Silicon Valley effect. At least one new company coalesced around and tried to exploit each new invention or discovery that came out of the lab. Notwithstanding spinoffs in all directions, there was still plenty to keep Fairchild Semiconductor growing as rapidly as it could. (Moore, 1996:167)

In 1965 there were 14 firms had been spinning-off semiconductor firms in Silicon Valley.[6] By 1970 the number had grown to 35, by 1975 to fifty four, by 1980 sixty three, and in 1986 there were 102 “parent” companies involved in semiconductor spin-offs and start-ups in the Silicon Valley area (Assimakopoulos, Everton, & Tsutsui, 2003). In total, 129 semiconductor companies were launched in Silicon Valley between 1947 and 1986 (Castilla, Hwang, Granovetter, & Granovetter, 2000).

As Saxenian (1991; 1994) and others have noted, the interdependencies in the Silicon Valley semiconductor industry are numerous. This can be seen, for example, by looking the graphical representation of co-foundation, shown in Figure 1. In the figure, a node represents a person who has founded a semiconductor company in Silicon Valley and two nodes are linked if the persons have been involved in founding the same company.

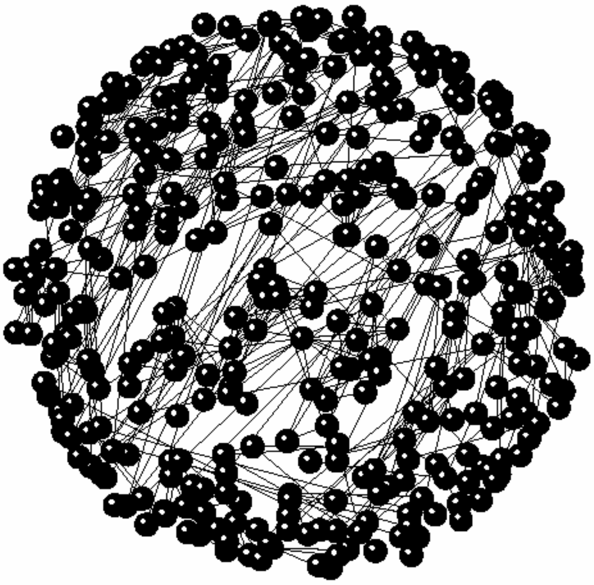

Figure 1. Co-founders in the Silicon Valley semiconductor firms (Castilla, Hwang, Granovetter, & Granovetter, 2000).

[6] The data comes from the Silicon Valley genealogy chart that shows links between semiconductor spin-offs. A firm is defined as a “parent” of a new firm if a founder of the new firm was employed by the parent of if the parent firm was one of the founders. I have not been able to gather accurate historical information about the nature of the new firms, i.e., to what extent they were real spin-offs or, for example, joint ventures of existing firms. In this definition, the parent firms are the “donors” of founders, and it is not always clear that the parent firm should be considered as a source of spin-offs. For example, Standard Oil appears as one of the parent firms of the Silicon Valley semiconductor industry, which probably does not imply that Standard Oil would in any conventional sense be a source of semiconductor spin-offs.

Dynamics of innovation in the semiconductor industry

Silicon Valley region became a center of ICT industry through a process of organic growth. The factors that have supported and facilitated this growth have varied across time. For example, the institutional and economic structures that enabled the rapid expansion of Internet-based businesses in the second half of the 1990s were to a large extent non-existent during the first decades of the Silicon Valley development. Undoubtedly, Silicon Valley and the broader San Francisco Bay region were fertile grounds for venture creation in the 1990s. When we try to understand whether similar dynamics of growth could occur in other settings, the early history of Silicon Valley is, however, in a crucial position. To find the generative factors that underlie the growth path of Silicon Valley we need both to understand its early history and the reasons why specific growth factors have been central.

It seems that the rapid growth of the semiconductor industry—and more broadly ICT industries—in Silicon Valley results from two important factors. First, semiconductor technology has seen extremely rapid but incremental improvements for almost half a century. In this sense, semiconductor products form a historically perhaps unique product category. The underlying technology trajectory has followed a continuous path since the invention of the planar process in 1957, where incremental innovation has been driving development and where discontinuous change has not destroyed competences (Tuomi, 2002). Although semiconductor industry is the prototypical innovation-based high-tech industry, it has seen limited creative destruction. This has provided opportunities to apply specific knowledge in closely related businesses.

Second, the industrial structure that has formed around this core technology has been based on spin-offs where new firms organically emerge from existing ones. This has led to extensively connected social networks, where shared histories have facilitated sharing of information, generation of knowledge about specific capabilities of people and institutions, and interpersonal trust.

I have argued elsewhere (Tuomi, 2002) that two different modes of innovation have been important for the historical development of ICTs. One is based on organic specialization of knowledge and spin-off of developer communities from a shared root community. This process creates technologies, communities, and knowledge that are historically connected. As the different communities share many practices and systems of value and meaning, knowledge moves easily across the developer communities.

Another form of development is based on a combinatorial dynamic. Technologies and knowledge that have been created in relatively independent communities and for the needs of independent social practices can be brought together to solve a practical problem. The communities that have developed the elements that have to be combined to solve the problem may not necessarily see a problem in a similar way, and often their knowledge and material outputs are appropriated in unanticipated ways. The history of technology has many examples of this process, including the birth of turbojet engines (Constant, 1980) and modern plastics (Bijker, 1997). To some extent, this combinatorial logic of innovation also has been reflected in product development approaches that emphasize modularity and interdisciplinary design practices.[7]

The interactions between these two different innovation dynamics have lead to co-evolution of technical architectures and social differentiation. The underlying social and technical structures hide complexity by standardizing interfaces and by defining "passage points" that control development.

From a knowledge and competence creation perspective, the organic model of technology development has special interest. Spin-offs create an ecology of historically interlinked stocks of knowledge and practices that facilitate social learning. Historically shared collaborative roots also mean that the social diversification of community structure leads to an ecology of sub-communities where the members often have strong social and cognitive ties. In linguistic terms, such sub-communities speak dialects of the same language.

An example of the evolution of the community structure in the history of the Internet can be seen in Figure 2 below. It starts from the famous Network Working Group, set up in 1968, that consisted of about a dozen computer students who had the task of defining applications for the ARPANET. During the history of the Internet, this group spun-off a large number of informal and formal communities that actually created the most important technologies and applications of the Internet.

A similar organic model of technology development and division of labor can be seen in large open source software communities. Projects such as the Linux operating system kernel development project have spun-off a large number of sub-projects that develop relatively independent modules of the system.[8] Coordination between these sub-projects is possible in part because historically they share the same genetic roots. As a result, even when the communities organize their work around different tasks and goals, they have historically been socialized in common practices and value systems that enable efficient collaboration and communication between different sub-communities.

[7] Product modularity, however, is also often associated with reuse, product variation strategies, and the need to create process and product flexibility in rapidly changing competitive environments. The combinatorial innovation dynamic discussed here is essentially a demand-side phenomenon, where different social practices are recombined in ways that generate new social practices (cf. Tuomi, 2001; 2002; 2003). The supply side modularity has been described by Baldwin and Clark (2000). *Design Rules, Vol 1: The Power of Modularity*. 2000. MIT Press, and knowledge integration, among others, by Kodama (1995).

[8] For a detailed discussion on the Linux kernel development model and the evolution of social and technical interfaces, see Tuomi, 2001 and Tuomi, 2002: Ch. 10.

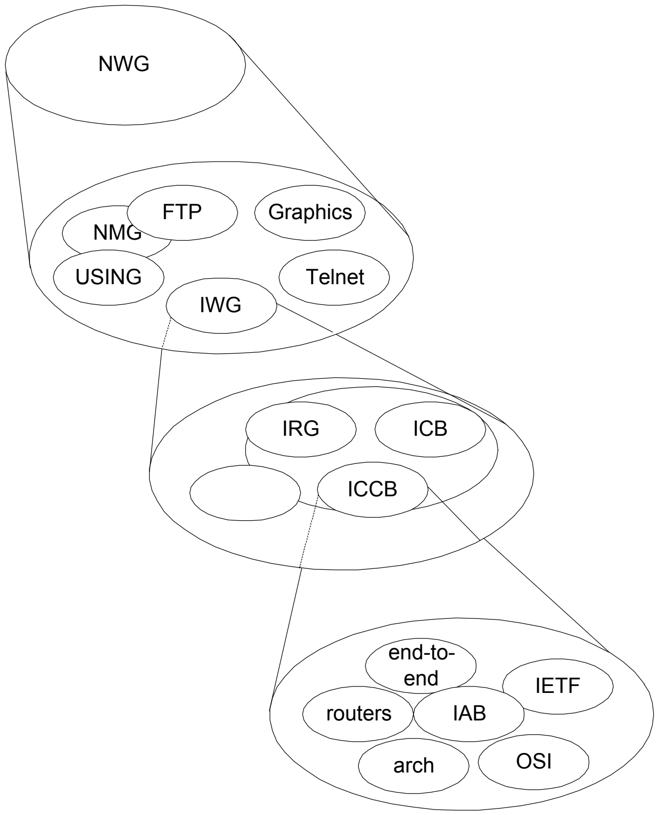

Figure 2. Production of fractal spin-off community structure (Tuomi, 2002: 151).

Weak intellectual property rights and spin-off firms

In the organic model of technology evolution, the core technologies of the technical trajectory are applied in many different areas, and the associated developer communities organize their development around increasingly diversified practices. Eventually, the evolution of socio-technical specialization may lead to world views and "thought communities" (Fleck, 1979) that are quite independent, incompatible, and incomprehensible to each other. When the historical roots are relatively proximate and within individual life span, diversification, however, leads to variations that the players still can recognize as modifications of the original theme. In this setting, knowledge flows easily from one community to another. Practices are similar enough that their differences make sense. Shared history and value systems mean that trust can be based on shared behavioral expectations and historical track-records. Institutional systems of trust can be bypassed and direct social links can provide access to social, economic, and cognitive resources.

The spin-off mechanism, however, only works if people are allowed to use accumulated knowledge in ways that are often in direct competition with the interests of the institutional parents who provided the setting and resources for its accumulation. For example, if intellectual property policies would have been enforced in Silicon Valley in ways that would have made competing spin-offs impossible, the spin-off dynamic could never have created regional economic growth. Historically, competencies and knowledge created in the parent firms, however, have quickly been put into new and often competing uses in the spin-offs. From the point of view of the parent firms, the Silicon Valley model therefore appears as a leaky bucket. The growth of the region has been made possible by a continuous flow of knowledge that has created a rich ecology of related businesses and allowed them to bloom.

The dynamic of spin-offs in Silicon Valley has been possible in part because of the limited use intellectual property rights in the semiconductor industry. Many key innovations for integrated circuits and computing systems were not patented. Knowledge about new technologies moved relatively freely across firm boundaries. Fairchild, for example, never patented ROM or RAM circuits, and Intel did not patent the microprocessor. To quote Gordon Moore:

> Well, it was probably a different attitude about patents. One thing that happened in the semiconductor industry... semiconductor processes are a long series of steps and the patents had gotten pretty broadly spread because all of the people working on the technology had some of them. And the net result was in order for any of us to operate we had to be cross-licensed so the participants tended to all cross-license one another. So, there was not a tremendous advantage to having more patents... with a couple of exceptions, there wasn't much net benefit from it.
>
> What we never anticipated, I guess, was a lot of other participants were going to enter the business later on. So, at Fairchild we tended to patent relatively few things, typically the ones that we thought we could police most easily and were the most difficult to get around, you know, the more fundamental things. But, I was responsible for a lot of those decisions. I remember one in particular that, in retrospect, is kind of funny. In the early days of the integrated circuit, Bob Norman, one of the people who were involved there, suggested the idea of semiconductor memory... the whole idea of how semiconductor flip-flops could be used as a memory structure, and I decided it was so economically ridiculous, it didn't make any sense to file a patent on it.
>
> …Well, you ask about patenting on the microprocessor and frankly, we didn't think the microprocessor *per se* was that patentable. What we had done was take a computer architecture and make it all on one chip instead of on several chips. And that was kind of the direction that the integrated circuit technology was pushing in anyhow, always putting more and more of the system on a chip. What TI did was then start saying: 'Well, a microprocessor with a keyboard is an 'invention,' and I'll admit, I never would have thought of filing patents on those things that TI got issued patents for. (Moore, 1995)

In summary, the dynamic of spin-off entrepreneurs has been facilitated by limited protection of intellectual property. Spin-offs were often created in Silicon Valley using knowledge that was created in their parent companies. Without the possibility to legally utilize knowledge that had been accumulated in the parent company, a spin-off would have faced great difficulties. The early dynamic of Silicon Valley was to a large extent based on entrepreneurial and innovative uses of shared knowledge.[9]

[9] One should, however, emphasize that the IPR regimes in the semiconductor industry have varied considerably across time. Intel, for example, was quite aggressive in its use of patents to create entry barriers already in the 1970s. I’m grateful for Paul Duguid for reminding me of the early patent fights in the Silicon Valley semiconductor industry.

Historically, shared knowledge has played an important role in the development of ICT industry also more broadly. Already in the 1940s, AT&T decided to widely distribute knowledge about transistor technology, as it was not able to efficiently commercialize it due to U.S. antitrust policies.[10] As a part of their integrated circuit patent dispute settlement in 1966, Fairchild and Texas Instruments agreed to license their IC technology to anyone with 2 to 3 percent licensing fee and (Langlois & Steinmueller, 1999). As the quote from Moore above points out, much of the IC-related knowledge was not patented. When firms had sufficient number of patents for cross licensing, there did not seem to be strong incentives to file more patents.

Broad access to technical knowledge and weak protection of intellectual property rights apparently has been important in the development of the semiconductor industry.[11] Also more broadly, a possible hypothesis could be that weak intellectual property rights have facilitated rapid growth in other ICT industries, the development of the Internet being perhaps the most prominent example.[12]

Intellectual property rights policies shape the spin-off process in an important way. When IPR policies do not strongly constrain the flow of knowledge from parent communities to its spin-offs, new specialized communities can be easily formed. If, on the other hand, strong IPR regime is in place and controlled by the parent community, new spin-offs can emerge only by an explicit agreement with the parent. In other words, a weak IPR regime is conductive for the innovative dynamic where spin-offs are important. Although the IPR regime is only one factor in the shaping of the developmental path and there can be many indirect effects that operate in parallel, a reasonable first approximation might be that strong IPR regime slows down the entry of competitive spin-offs.[13]

It appears that the economic development of Silicon Valley was dominated by this spin-off dynamic during the first three decades of its history. The spin-offs did not only create semiconductor firms, but also a rich ecology of complementary enterprises and institutions. For example, studies on the history of venture capital industry (Florida & Kenney, 1988; Kenney & Florida, 2000; Wilson, 1985) show that many of the Silicon Valley venture capital firms have been closely related to the

[10] Apparently, one reason was also that AT&T believed that it was useful to spread this potentially important technology quickly so that that it would develop faster and become usable in AT&T's mainstream products, and also because rapid dissemination of the underlying ideas would lower the probability that military would declare it as a military secret.

[11] The "weakness" of intellectual property rights refers here to the strength by which they constrain non-owners. For example, the GPL open source license is in this sense "weak" as it does not strongly constrain the actions of non-owners. The GPL copyright, of course, is as strong as any other copyright and it is exactly this strength of copyrights that makes open source possible. From the point of view of growth dynamics, the impact of intellectual property rights does not depend on abstract characteristics, but on their concrete use. In this sense, although the U.S. and the EU IPR systems are legally quite similar, they differ greatly because the opportunities and costs to use IPR to create competitive constraints differ considerably.

[12] RAND published widely its work on packet switching that later provided an important foundation for the development of the Internet. Due to antitrust policies, AT&T also provided free access to the c-programming language and Unix in the 1970s, which laid the foundation for the growth of the Internet. The World Wide Web has also been developed under weak intellectual property rights.

[13] For example, strong patent rights may be important for gaining access to venture capital and knowledge networks managed by venture capitalists (cf. Hall & Ziedonis, 2001).

semiconductor industry.[14] As Kenney has argued, venture capital firms have played an important role both in matching people and firms, and in the management of knowledge flows. In Nonaka's (1988) terms, venture capitalists have acted as "middle managers" linking knowledge horizontally and mobilizing resources than can and need to be mobilized.

Patenting policies in the semiconductor industry changed in the 1980s (Hall & Ziedonis, 2001). One may, however, argue that the foundations of Silicon Valley, as an innovative R&D intensive region, were already fully in place at this time, and that the current strong patent protection policies have not been important for the ICT industry growth in Silicon Valley.[15] As a result of the strengthening of the IPR regime in semiconductors in the 1990s, patent portfolios become increasingly important for cross-licensing. The dynamic impact of strong patent protection was perhaps mainly visible in the slowing down competitors in an industry where system compatibility and first-mover advantages are critical.

In summary, therefore, it is possible that the current IPR regime would have been counterproductive for venture creation and regional economic growth during the first two and half decades of the Silicon Valley history. In other words, Silicon Valley could be a historically unique event that could not be repeated under current conditions. If someone would today invent technologies similar to planar transistor and integrated circuit, the development would perhaps be much slower than what it historically was in the semiconductor industry.

Software and industry growth

One may ask whether the developmental dynamic of semiconductors could have materialized in other geographical regions. If Shockley had started his laboratory somewhere else in 1956, would this place now be called Silicon X? Did the weak intellectual property regime and the inherent technical possibilities of silicon provide the sufficient foundation for launching the growth process that eventually became the Silicon Valley? Could a weak IPR regime and another knowledge-intensive product category lead to growth processes that would resemble what we have seen in the Valley?

Other factors, of course, have been important for the development of what we today know as the Silicon Valley. These include the development of computer and software industries, and the role of institutions such as Xerox PARC.[16] Yet, one may ask

[14] For example, Kleiner Perkins Caufield & Byers and Sequoia Capital have their roots in Fairchild.

[15] In fact, the current strong patent protection policy in the semiconductor industry was initiated by Texas Instruments, perhaps because it had to compensate its competitive disadvantage in accessing the Silicon Valley innovation system. Software patents, of course, were important in the last years of the 1990s. Their importance, however, probably was to a large extent symbolic, as they were viewed as a condition for getting venture capital. In a sense, software patents were used as poison pills that made it expensive for competitors to slow down firm growth by binding managerial and financial resources in lawsuits.

[16] The location of Xerox PARC was partly influenced by the business friendly attitude of Stanford and availability of cheap rental estate. The existence of semiconductor or computer firms in the region did not seem to be of major importance. The factors that were considered in the selection process have been described by Hiltzik (2000:36-8). Hewlett-Packard apparently started to become directly connected with the development of the semiconductor industry only in the 1970s.

whether other technologies, for example, biotechnology, nanotechnology, new materials, or communications technologies could produce similar growth patterns. In the following, I briefly focus on software technologies and the European software industry. I start by looking the growth process that underlies the development of the GNU/Linux operating system.

One well-known example of rapid growth in software is the Linux operating system kernel. Although in operating system development less code is often better, and typically nine-tenths of the source code are removed during system development in open source projects, the Linux kernel has been growing rapidly since 1991. This can be seen in Figure 3 below.

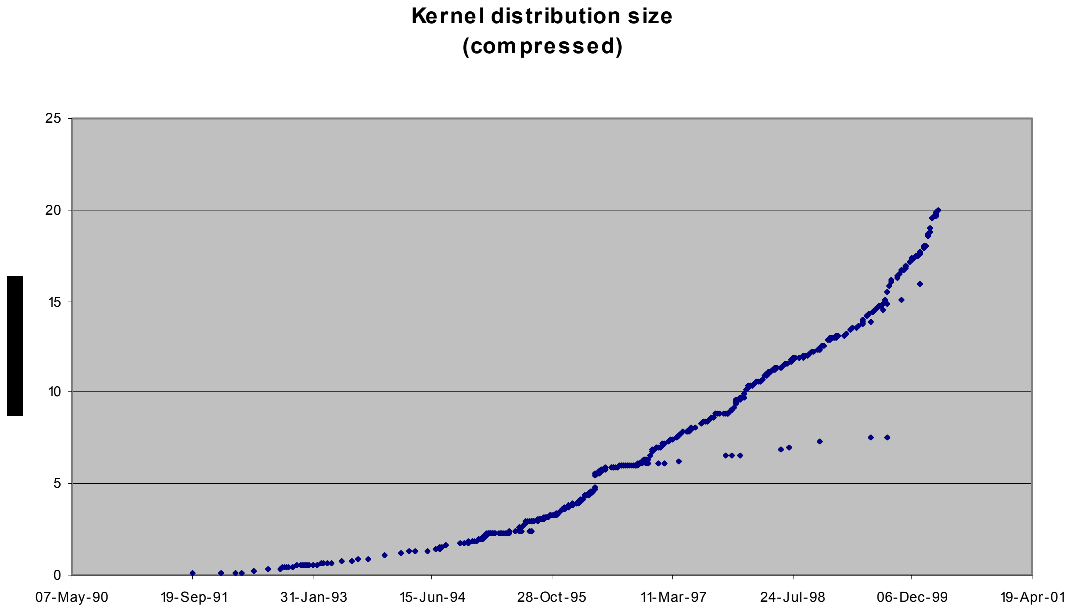

Figure 3. Linux operating system kernel size (Tuomi, 2002: 167).

The rapid growth in the Linux kernel has been associated with simultaneous modularization of the code architecture and specialization of the sub-communities that develop the system. These spin-offs have often emerged from a broader software developer community, but the open source development model has facilitated socialization of new developers into the core kernel development community. A more detailed analysis of the historical Linux development model shows that the rapid growth of the source code has been concentrated in areas that complement and augment the functionalities of the core operating system. Openness and weak intellectual property rights have been critical both for the creation of development competences and integration of knowledge across different sub-communities.

In this sense, the Linux development model and the early history of semiconductor industry can be argued to be similar.[17] The "virtual" innovation region of Linux development has created a strongly connected cluster of related development practices

[17] There are also differences, for example, as the focus on Linux implicitly requires that we forget the competing spin-offs, i.e., forking of code.

and competences. This virtual region, however, is globally distributed and no physical location has emerged as a center of development (Tuomi, 2004).

Open source projects such as Linux are, however, quite special in the way they use intellectual property rights. In general, software industries develop in contexts where intellectual property regimes and other factors vary. For example, Baba et al. (1996) show that the Japanese software industry has to a large extent been based on spin-offs from user industries, such as the steel and banking industries, where knowledge about both computing technology and its applications were simultaneously present. To a lesser extent the Japanese software industry consists of spin-offs from computer manufacturing industry, and independent software houses represent only a relatively small fraction of the industry. This dynamic depends on the particular way Japanese economy and society is organized. Merges (1996) argued that strong intellectual property was not important in Japan, as most software development has been done under explicit or implicit contracts between software suppliers and their often large customers.[18] In Japan, in other words, the industry structure has to an important extent influenced the intellectual property regime. This developmental path, however, has become a dead-end, as the competitive advantages of standards and low transaction costs in the packaged software market have started to drive growth. Merges therefore also argued that too weak intellectual property rights can slow down growth and innovation in software industry, and that the growth in Japanese packaged software industry has, in fact, been slowed down because of this.

Due to the potentially large network effects, lock-in, path dependencies, and cost dynamics in software production, it has been speculated that the software industry easily produces natural monopolies (Mowery, 1996). In such a context, natural and IPR monopolies may amplify each other's impact. The structure of European software industry, for example, shows that the growth dynamic in software products and services apparently leads to a concentrated industry structure. This can be seen in Figure 4, which shows the number of active software consultancy and supply firms in eight EU countries, categorized using the number of employees.

[18] Custom software represented about 85 percent of Japan's computer software market in 1997 (Anchordoguy, 2000).

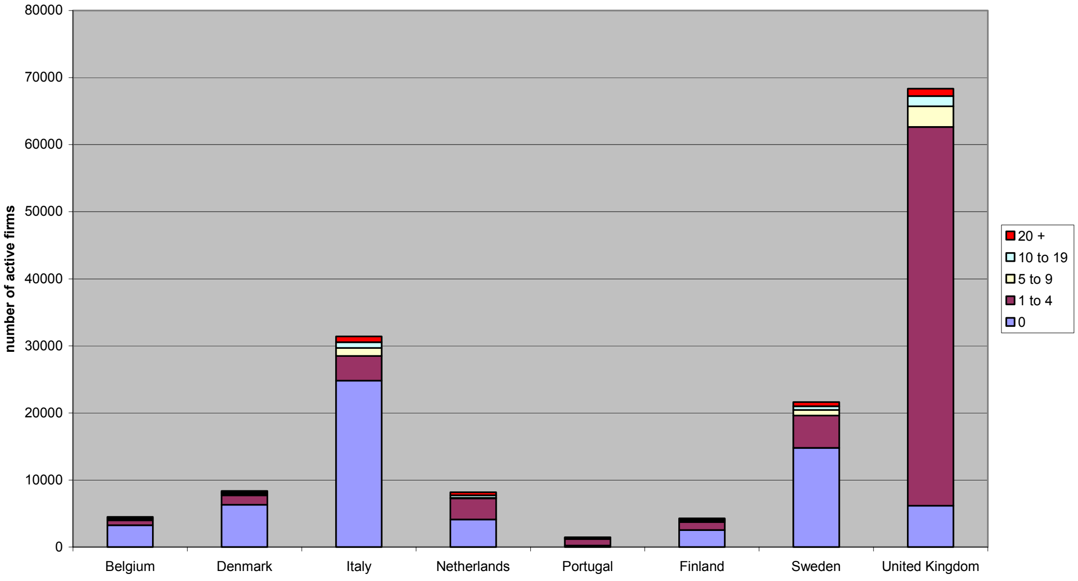

Figure 4. Number of employees in software consultancy and supply firms in some EU countries, year 2000.

The industry structure is more clearly visible in Figure 5. It shows the total number of firms in the eight EU countries, categorized by the number of employees in year 2000. As can be seen, a large majority of software firms have less than five employees.

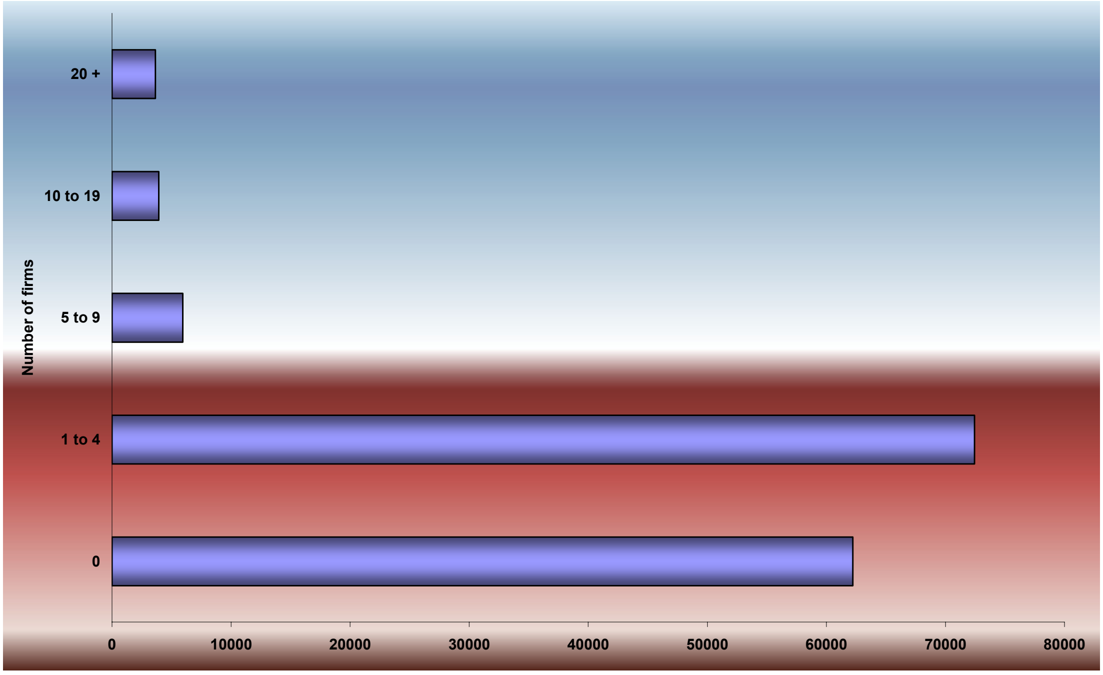

Figure 5. Number of active software consultancy and supply firms in eight EU countries, year 2000.

Although the number of firms and concentration of software industry increased during the second half of the 1990s, the proportion of firms with over twenty employees has remained stable. This can be seen Figure 6. Over 90 percent of all software consultancy and supply firms had four employees or less. About 2.5 percent of the firms had 20 employees or more.

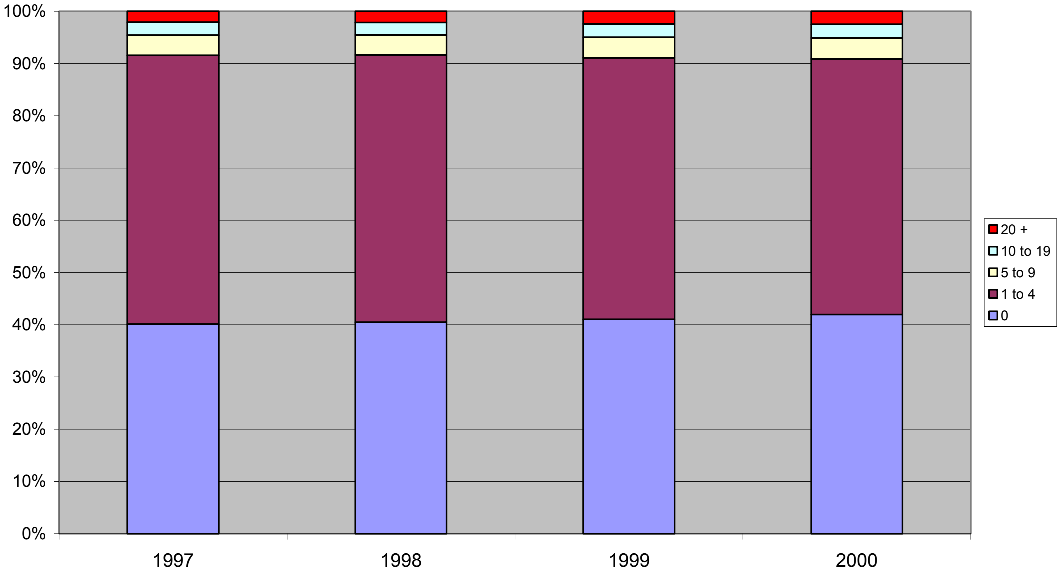

Figure 6. Software consultancy and supply firms in eight EU countries.

The firm size distributions vary across countries. Italy, for example, has a large number of active software firms without employees, whereas the UK has few firms with zero employees but many firms with less than five employees. The aggregate number of zero to four employee firms, however, is relatively similar across the eight EU countries for which data is available. In Italy, firms with more than five employees represent ten percent of all software firms and in the UK they represent eight percent.

The figures above provide a hint of the phenomenon that we are looking for. Continuous innovation in software industries requires a constant flow of competing technologies that in principle could be created by new entrants.[19] To simultaneously create innovation and growth in the economy, small firms need to grow.[20] To access the "Silicon Valleys" of software industry, new entrants have to cross death valleys where new ideas and the capabilities to make them market realities are only distant mirages. Apparently, few firms have been able to do this in Europe. Most medium-sized and large software consultancies and suppliers in Europe focus on customized software production, where outsourcing creates increasing competitive pressures. The innovation pipeline may currently be an important bottleneck for industry growth.

[19] The question whether large or small firms drive innovation is, of course, one of the key issues in innovation research. The Schumpeterian innovation model, for example, can be interpreted both to argue that an increasing amount of R&D occurs in large corporations, or that new entrants create the new domains of economic activity that make growth possible.

[20] This argument need some further justification, which can be found from research on the role of dominant designs, standards and interoperability, internal venturing, and incentive dynamics in software industries. In essence, I will argue that dominant players have so big incentives for customer lock-in that disruptive innovation is too disruptive, unless radical new technologies emerge that potentially destroy the market segment. If the industry profits depend on natural monopolies, product cannibalization rarely makes sense.

Conclusion

The data shown above should be disaggregated to substantiate claims about the dominance of specific developmental dynamics in the software industry. In general, software industry produces different types of outputs that have different growth paths. Too broad aggregation, therefore, easily misses important factors of development. It is therefore important to note that the discussion above is speculative and intended only to promote discussion, thinking and future research.

Software industry is often segmented to system-level software, tools, and applications producers. The growth factors of system-level software most probably differ from those of application level software. The industry also sells its outputs in various forms, including packaged software, custom developed application software and outsourcing services.[21] In many countries, the biggest software producers are not counted as software producers, as they develop software that is embedded in material products. In general, accurate data about software production are difficult to find and often cannot easily be compared across countries or regions. Yet, it is also clear that software is one of the most important industries in the knowledge society. It is therefore useful to develop improved data and models about software production. Although it may be impossible to say today how, for example, intellectual property policies should be implemented to remove obstacles for growth in the software industry, at least we can say that the question is relevant.

This incomplete working paper has tried to make some first steps in that direction. I have put forward some ideas and made some claims that need to be justified. One underlying claim, however, should be relatively uncontroversial. By studying history, we can better understand development and growth.

[21] The global software industry has grown to an important extent during the last years as a result of international outsourcing, which is not always easily captured in national industry statistics. For example, Arora and Athreye (2002) point out that out of Fortune 500 companies, 185 outsourced their software production to India in 2002.

References

Anchordoguy, M. (2000). Japan's software industry: a failure of institutions? *Research Policy,* **29**, pp.391-408.

Arora, A., & Athreye, S. (2002). The software industry and India's economic development. *Information Economics and Policy,* **14**, pp.253-73.

Assimakopoulos, D., Everton, S., & Tsutsui, K. (2003). The semiconductor community in the Silicon Valley: a network analysis of the SEMI genealogy chart (1947-1986). *International Journal of Technology Management,* **25** *(1/2),* pp.181-99.

Baba, Y., Takai, S., & Mizuta, Y. (1996). The user-driven evolution of the Japanese software industry: the case of customized software for mainframes. In D.C. Mowery (Ed.), *The International Computer Software Industry: A Comparative Study of Industry Evolution and Structure.* (pp. 104-130). New York: Oxford University Press.

Baldwin, C.Y., & K.B. Clark. (2000). *Design Rules - Vol. 1: The Power of Modularity.* Cambridge, MA: The MIT Press.

Bijker, W.E. (1997). *Of Bicycles, Bakelites, and Bulbs: Toward a Theory of Sociotechnical Change.* Cambridge, MA: The MIT Press.

Castells, M., & P. Hall. (1994). *Technopoles of the World: The Making of 21st Century Industrial Complexes.* London: Routledge.

Castilla, E.J., Hwang, H., Granovetter, E., & Granovetter, M. (2000). Social networks in Silicon Valley. In C.-M. Lee, W.F. Miller, M.G. Hancock, & H.S. Rowen (Eds.), *The Silicon Valley Edge: A Habitat for Innovation and Entrepreneurship.* (pp. 218-247). Stanford, CA: Stanford University Press.

Constant, E.W. (1980). *The Origins of the Turbojet Revolution.* Baltimore: Johns Hopkins University Press.

Fleck, L. (1979). *Genesis and Development of a Scientific Fact.* Chicago, IL: The University of Chicago Press.

Florida, R., & Kenney, M. (1988). Venture capital, high technology and regional development. *Regional Studies,* **22** *(1),* pp.33-48.

Hall, B.H., & Ziedonis, R.H. (2001). The patent paradox revisited: an empirical study of patenting in the U.S. semiconductor industry, 1979-1995. *RAND Journal of Economics,* **32** *(1),* pp.101-28.

Hiltzik, M.A. (2000). *Dealers of Lightning: Xerox PARC and the Dawn of the Computer Age.* New York: HarperCollins Publishers.

Kenney, M. (2000). *Understanding Silicon Valley: The Anatomy of an Entrepreneurial Region.* Stanford, CA: Stanford University Press.

Kenney, M., & Florida, R. (2000). Venture capital in Silicon Valley: Fueling new firm formation. In M. Kenney (Ed.), *Understanding Silicon Valley: Anatomy of an Entrepreneurial Region.* (pp. 98-123). Stanford, CA: Stanford University Press.

Kodama, F. (1995). *Emerging Patterns of Innovation: Sources of Japan's Technological Edge.* Boston, MA: Harvard Business School Press.

Langlois, R.N., & Steinmueller, W.E. (1999). The evolution of competitive advantage in the worldwide semiconductor industry, 1947-1996. In D.C. Mowery & R.R. Nelson (Eds.), *Sources of Industrial Leadership: Studies of Seven Industries.* (pp. 19-78). Cambridge: Cambridge University Press.

Merges, R.P. (1996). A comparative look at intellectual property rights and the software industry. In D.C. Mowery (Ed.), *The International Computer Software Industry: A Comparative Study of Industry Evolution and Structure.* (pp. 272-303). New York: Oxford University Press.

Moore, G.E. (1995). Interview with Gordon E. Moore, March 3, 1995. Silicon Genesis: Oral Histories of Semiconductor Industry Pioneers. Program in History and Philosophy of Science, Department of History, Stanford University

Moore, G.E. (1996). Some personal perspectives on research in the semiconductor industry. In R.S. Rosenbloom & W.J. Spencer (Eds.), *Engines of Innovation: U.S. Industrial Research at the End of an Era.* (pp. 165-174). Boston, MA: Harvard Business School Press.

Mowery, D.C. (1996). *The International Computer Software Industry: A Comparative Study of Industry Evolution and Structure.* New York: Oxford University Press.

Nonaka, I. (1988). Speeding organizational information creation: toward middle-up-down management. *Sloan Management Review, (Spring),* pp.57-73.

Saxenian, A. (1991). The origins and dynamics of production networks in Silicon Valley. *Research Policy,* **20** *(5),* pp.423-37.

Saxenian, A. (1994). *Regional Advantage : Culture and Competition in Silicon Valley and Route 128* . Cambridge, MA: Harvard University Press.

Saxenian, A.L. (1999). *Silicon Valley's New Immigrant Entrepreneurs.* San Francisco: Public Policy Institute of California.

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

Tuomi, I. (2002). *Networks of Innovation: Change and Meaning in the Age of Internet.* Oxford: Oxford University Press.

Tuomi, I. (2002). The lives and death of Moore's Law. *First Monday,* **7** *(11)* http://firstmonday.org/issues/issue7_11/tuomi

Tuomi, I. (2003). Beyond user-centric models of product creation. User Aspects of ICT: The Good, Bad and the Irrelevant. Helsinki 3-5, September 2003.

Tuomi, I. (2004). Evolution of the Linux Credits file: methodological challenges and reference data for open source research. *First Monday,* **9** *(6 (June))*

Wilson, J.W. (1985). *The New Venturers: Inside the High-Stakes World of Venture Capital.* Reading, MA: Addison-Wesley Publishing Company.