A Danish view on software-related patents*

Ulrich Kaiser and Thomas Rønde*****

Abstract: This paper analyzes the likely effect of changes in European patent legislation on the Danish economy. We review the relevant literature on patents in general and on software-related patents in particular. We also illuminate the role that software-related inventions play in Denmark. Our main policy conclusion is that software patents are less likely to be beneficial for Denmark than for the US since Denmark lacks a significant software industry. At the same time Denmark is a significant adopter of information and communication technologies.

Keywords: software patents, information and communication technology, Denmark
JEL classification: 034

---

* This research greatly benefited from positive research spillovers received from the following individuals: Joachim Henkel (Munich School of Business, Ludwig-Maximilians-University, Munich), Michael Kölling (Mærsk-McKinney Møller Institute for Production Technology at the University of Southern Denmark at Odense), Georg Licht (Centre for European Economic Research, Mannheim) and Kurt Crone Jørgensen (Grundfos A/S, Bjerringbro). Helpful input was also received from Stefan Wagner (Munich School of Business, Ludwig-Maximilians-University, Munich). We gratefully acknowledge partial funding of this research from the Danish Ministry of Economic and Business Affairs. The views expressed here are those of the authors and do not necessarily represent the views of the ministry. The usual disclaimer applies.
** Ulrich Kaiser: Dept. of Economics, University of Southern Denmark at Odense, Campusvej 55, DK-5230 Odense M, email: uka@sam.sdu.dk, internet: http://www.sam.sdu.dk/staff/uka; Centre for European Economic Research, Mannheim, and Centre for Economics and Business Research, Copenhagen.
*** Thomas Rønde: Dept. of Economics, University of Copenhagen, Studiestræde 6, DK-1455 Copenhagen K, email: Thomas.Ronde@econ.ku.dk, internet: http://www.econ.ku.dk/staff/new/profile.asp?ID=1814, Centre for Economic Policy Research, London, Centre for Industrial Economics, Copenhagen, and Centre for Economic and Business Research, Copenhagen.

1. What is at stake?

A spotlight has been cast on software patents in September 2003 when the European Parliament – under heightened media attention – first delayed a vote on a controversial European software patent proposal and then amended it in a way that even sharp critics came to agree.

The patent directive is now, in early 2004, with the European Commission for review. Votes in parliament and the Council of Ministers will follow. If approved, it will be implemented in the national laws of the EU member states. The changes made to the patent directive by the European Parliament induced EU Commissioner Frits Bolkestein to threaten to withdraw the proposal in favor of negotiating an intergovernmental treaty that would not require any voting – a move that would imply a lot of space for intervention by national EU governments. The earliest date from which the EU directive will take effect, without intervention by the EU commission, would be the end of 2005.

What had happened and why are software patents such a controversial topic? In an initiative of the European parliament's Committee on Legal Affairs and the Internal Market (JURI), a new software patent law was developed that would have moved European legislature closer to the US practice. US and European software patent legislature could not have been more different: in the US there basically do not exist limits to software patentability, at least since 1995 ("State Street Bank" case), while in Europe only "computer implemented inventions" are patentable. The invention must bear "technicity" and be in connection to a technological application. There hence is no patenting of "software as such", basically meaning that "pure" software code that does not come with a technical application is patentable. In Europe, only software-related inventions that have a "technical character" are patentable, which means that the invention has to perform a task through an apparatus or that it has to have an influence on the technical properties of the apparatus. The notion of "technicity" has led to great confusion among business and patent examiners alike. As a consequence, many patents have been granted to software-related inventions in Europe and Denmark, and much of a patent application's success depended on a lucky formulation of the application text. Examples of software-related inventions that were granted patent protection by the European Patent Office (EPO) are inventions related to steering- and regulation technologies, digital signal processing technologies, hardware drivers, PC operation systems and Computer Aided Design/ Computer Aided Manufacturing-related innovations. Patents have, however, not been granted on business processes.

The introduction of software-related patents in the US has not been without problems and has led to many seemingly trivial patents such as to the well-known one-click-only online shopping patent granted to Amazon.com. These trivial patents are also the reason for recent discussions about the patentability of business methods in the US. Some observers believe that it is possible that the US Supreme Court restricts the patentability of software-related patents, in particular that of business methods (Blind et al. 2001).

The Japanese approach to the patenting of software-related inventions is the same as that of the US Patent and Trademark Office (USPTO): there are no limits to patenting. European countries are hence quite isolated with respect to their refusal to make computer-related inventions patentable. The proposed initiative by JURI led to an uproar that culminated in protest marches and virtual protests and eventually led the European Parliament to take the stings out of the patent law proposal.[1]

The patent law directive that eventually was approved by the European parliament does allow for software patents that are "true inventions" but disallows for patents on business methods such as one-click-only online shopping.

The aim of this paper is quite modest. In Section 2, we discuss the role of patents in ensuring innovation. In Section 3, we then proceed by exchanging arguments pro and contra a European software patent. Thereafter, we discuss in Section 4 the potential effects of a strengthened software patent law on Denmark. Finally, Section 5 contains policy recommendations and a discussion of the avenues for future research.

2. The Economics of Patents

2.1 The Traditional Role of Patents

In order to illustrate the role that the patent system plays for innovation suppose that a firm or an individual has an idea for an (non-obvious and useful) invention. A good idea, however, is not enough. The potential inventor needs to invest in research and development (R&D) to transform her idea into a marketable product or an improved

[1] Some observers, such as the UK-based IT news service "The Register" (The Register 2003) particularly appreciated the constructive role an open letter undersigned by 14 leading economists that work in the field of intellectual property rights, including Lee Davis of Copenhagen Business School and Bengt-Åke Lundvall of Aalborg University, played in the debate.

process. Suppose that an inventor decides to go ahead and develops the new product or process. If there were no patent laws, other firms could freely copy the new product. The rents that accrue to the inventor would therefore be reduced, possibly to the point where it would no longer be privately profitable to undertake the R&D project. Foreseeing this outcome, the potential inventor might choose not to develop her product in the first place. There are indeed cases where this happens. In India, a country with particularly weak patent laws, for example, the number of newly developed drugs is close to zero (while India is a major producer of generic drugs).

In order to prevent such an undesirable outcome, the patent system was introduced more than 150 years ago. Lerner (2002a and 2002b) gives an economic history overview. Today, an inventor can seek for patent protection of an invention and obtain a temporary, up to 20 years, monopoly on exploiting her invention. This prevents competitors from using the invention freely, and ensures the inventor a "healthy" (risk-adjusted) return on her R&D investment. The patent, however, comes with an obligation: the innovator has to disclose the nature of the invention publicly and in such detail that an expert in the field can replicate it. Other inventors can therefore use the knowledge embedded in the invention as input in their R&D. Costly duplication of R&D efforts is thus avoided. Put differently, a patent is a contract between the inventor and society where the inventor receives a temporary monopoly at the cost of disclosure. Figure 1 illustrates the basic economics of patents.

The patent system has both costs and benefits. On the one hand, it stimulates, at least in principle, innovation by securing a sufficient return on R&D investments and by ensuring that new knowledge is disseminated. On the other hand, it grants the inventor a monopoly over her invention, which reduces economic efficiency in the market. The optimal patent system has therefore to balance these costs and benefits. This is often described as a trade-off between *static efficiency* (competition) and *dynamic efficiency* (innovation). Stronger patent protection – whether in scope or length – stimulates innovation, because the return on R&D investments increases, but magnifies at the same time the inefficiencies caused by monopoly. From the point of view of overall welfare in society, the patent system should therefore ensure inventors a healthy return on their R&D investments without over-rewarding them.

Figure 1: *The functioning of the patent system*

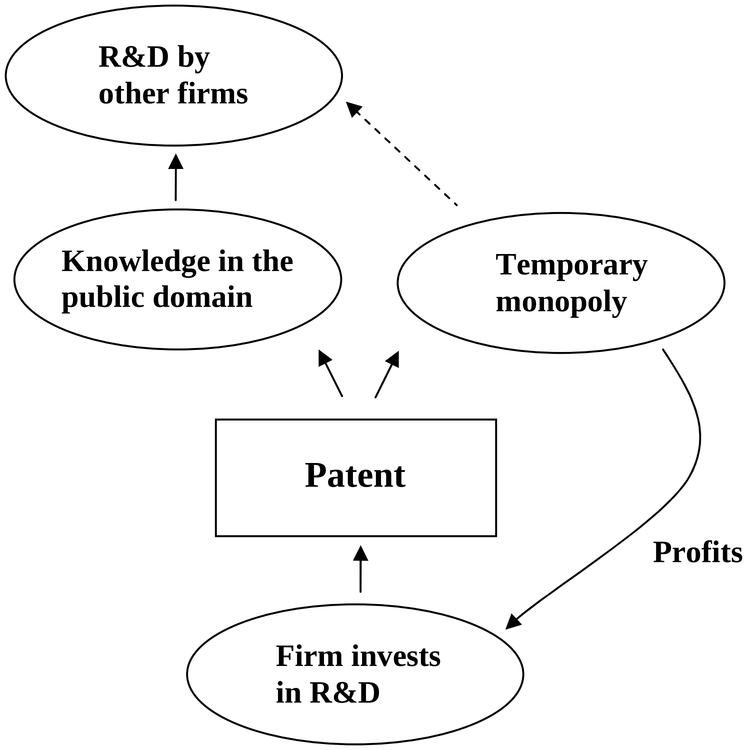

Figure 1 shows – in a simplified way – how the patent system works. Solid arrows are to be translated as "directly influences", the dashed arrow means "indirectly influences". Firms invest in R&D, this investment leads to an invention that is patented. The firm obtains a temporary monopoly on the invention and might also "hold up" other firms' R&D (dashed arrow). Once a patent application is filed, the invention is in the public domain and other firms get aware of it.

A well-designed patent system should in theory increase innovation, but the positive effects have been hard to measure empirically. In its argumentation in favor of software-related patents, the JURI committee claims that there is a link between R&D spending, patent applications, and productivity. This argument lacks, however, empirical basis. Empirical studies have not been able to establish a positive effect of patents on firm performance, that is above and beyond R&D spending (Griliches 1984; Hall 1996; Griliches et al. 1987). That is not to say that the patent system cannot have a positive effect on innovation in some industries, but it does suggest that a positive effect of software patents cannot be presumed.

There are also a number of other arguments that generally speak against the patent system. Business surveys in traditional industries have shown early that patenting is quite low on the list of firms' means to appropriate the returns from R&D (see Cohen et al.

2000 for the U.S., Arundel and Kabla 1998 for Europe and Blind 1999 for Germany). More important devices are secrecy, lead time advantages, and complementary assets (distribution and customer services).

2.2 The New Role of Patents in High-tech Industries

The effects of patents are even more controversial in high-tech industries where new issues arise. Indeed, most of the discussion concerning software patents centers around the special role that patents play in high-tech industries such as computer-related industries.[2] Before continuing, however, there is an important caveat that the reader should have in the back of the mind when reading this report: patenting behavior in high-tech industries is frontier economic research. All findings are preliminary and need to go through the process of critical assessment by the scientific community before robust conclusions emerge.

Innovations in high-tech industries are typically characterized by short product cycles where new and improved versions of the products replace the older ones. Innovation is therefore *sequential* as a new version of a product builds on the knowledge embedded in older versions. Furthermore, innovations are typically *complementary*: to produce a DVD player or a cell phone, it is necessary to combine inventions of many different firms. The combination of sequential and complementary innovations, and often strong network externalities,[3] give rise to very complex strategic situations. A firm that tries to develop a new product may easily end up infringing on an existing patent. Or, it may need some complementary technology, e.g. to produce a product that is compatible with the industry standard. The firm is in a weak bargaining position in such situations and is confronted with the risks of paying a high price for the license of the necessary technologies. This is known as a 'hold-up' situation in the economic literature.

Compared to the discussion in section 2.1, the new aspect here is that granting a patent to an inventor influences future innovation, because the inventor can hold-up future

[2] Note that most software patents in Europe and the US are granted to hardware manufacturers, not to software producers. For example, the number one software patent holder in Europe is IBM, followed by Siemens. By contrast, Microsoft is ranked 15th only.

[3] Network externalities arise when the benefit that a customer derives from a good is increasing in the number of other customers buying the good. For example, the more people use Windows as an operating system, the more Windows compatible programs will be developed, and the higher is the benefit from buying Windows. Likewise, the more widespread the use of PDF files is, the higher is the benefit from buying Acrobat Distiller to be able to create such files.

inventors that need the invention as an input. This is illustrated by the dashed arrow in Figure 1. There are two major channels through which hold-up problems affect future innovation. First, future inventors foresee the possibility of a hold-up which reduces their incentives to do R&D.[4] Second, once an inventor has started to develop a new product and would like to employ a patented technology, the negotiations concerning the licensing conditions may break down. This could cause the product to be abandoned altogether or at least impose significant costs on the inventor who has to find a different solution.[5] Hold-up problems may thus result both in *ex-ante* inefficiencies (some products are never developed) and *ex-post* inefficiencies (new products are abandoned or developed at higher costs).

Firms are, of course, aware of the danger of hold-up and try in different ways to prevent this from occurring. Since the beginning of the 1990s there has been a sharp increase in the number of patents - without a similar increase in R&D expenses - and many observers explain this as the result of strategic patenting behavior where firms "pile up" patents to be prepared for cross-licensing negotiations or patent infringement cases. Patents have in other words become "bargaining chips", (Kash and Kingston 2001). Hall and Ziedonis (2001) find strong support for such patenting behavior in the software industry. Strategic patenting, in turn, increases the probability of hold-up since more patents are issued, thereby creating what some authors term a "patent jungle" or a "patent minefield" (Shapiro 2001).

A recent German survey provides some evidence on the role of patents in the computer industry (Blind et al. 2001). It is found that formal protection strategies like patents are of least importance. The German survey also came to the conclusion that firms from the primary software sector take out patents as a protection device against imitation, which is

---

[4] This argument is not as straightforward as it seems at a first glance. The future inventors will foresee that they, in turn, can hold up their future inventors, which could restore the incentive to invest in R&D. Still, there is probably no doubt that the risk of hold-up can prevent some valuable inventions from being developed.

[5] A US judge ruled that Microsoft Corp.'s Internet Explorer Web browser infringed on a patent owned by Eolas Technologies Inc. and the University of California and ordered the company to pay $520.6 million in damages in January. The patent describes in part "a system allowing a user of a browser program (...) to access and execute an embedded program object," or small computer programs, often referred to as "applets" or "plug-ins." Hence, any web browser that provides more than just plain text infringes on that patent. Microsoft announced to get around the patent infringement by developing its own software for applets and plug-ins. Source: Computerworld (2004).

the role intended for patents. However, firms from the secondary sectors (for example hardware producers) apply for patents for "strategic" motives.[6]

In many markets, the major players have also created patent pools that allow firms to buy a bundle of complementary technologies. This reduces the hold-up problem, because the firm has to negotiate licensing fees with fewer parties. Patent pools have, on the other hand, caused antitrust concern since they may serve as a barrier to entry into the industry or restrict competition. Lerner and Tirole (2002a) provide a rigorous analysis of the competitive effects of patent pools, but more work is needed on this topic.

2.3 Existing software-related inventions in Denmark

There are – according to different sources – between 8,000 and 20,000 software-related patents in Europe.[7] Nevertheless, we are not aware of any software patent studies for Denmark or for Europe. To have a rough idea of how many software-related patents there are in Denmark, we have analyzed the Danish patent data bank (http://dk.espacenet.com/). We follow the definition of software patents by Graham and Mowery (2002a,b).[8] Graham and Mowery define software-related patents according to the patents' International Patent Classification (IPC) number, an identifier that locates patents into technological classes. Eleven main groups in the IPC scheme are picked by Graham and Mowery:

G06F	**Electric Digital Data Processing**
3/	Input arrangements for transferring data to be processed
5/	Methods or arrangements for data conversion
7/	Methods or arrangements for processing data
9/	Arrangements for programme control
11/	Error detection; Error correction; Monitoring
12/	Accessing, addressing or allocating within memory systems or architectures
13/	Interconnection of, or transfer of information or other signals
15/	Digital computers in general
G06K	**Recognition of data**
9/	Methods or arrangements for reading or recognizing printed or written characters
15/	Arrangements for producing a permanent visual presentation of the output data
H04L	**Electric Communication technique**
9/	Arrangements for secret secure communication

[6] In the German survey "strategic patenting" motives are basically all reasons other than those that the patent system was originally intended for.

[7] Ingo Kober for example, at that time president of the EPO, said in 1998 that: "In fact in the 20 years in which it has been in operation, the EPO has granted somewhere in the region of 20,000 software-related patents (…)."

[8] According to Hart and Wallsten (2003) the Graham and Mowery (2002a,b) definition of software-related inventions is more rigorous than the competing Bessen and Hunt (2003) definition.

Table 1 shows how many software-related inventions we find in the patent data files of the Danish Patent and Trademark Office (DKPTO) in the time period 1990 and 2003.[9] The figures include both granted patents and patent applications and refer to the year of application. Note that there are double counts in this table since some inventions are patented under two or more different IPC classifications. The main message from Table 1 is clear: there are very few software-related patents granted or applied for in Denmark.

In Appendix A we look a bit deeper into the software-related inventions for which patent protection is sought for or is granted. An important finding with respect to the fact that some observers believe that the patentability of software-related inventions would put small and medium sized firms (SMEs) at a disadvantage, is that there are almost as many software-related inventions patented by private persons as there are by corporations.[10]

Table 1: *Patents and patent applications by Danish firms, by IPC number*

	G06F3/	G06F7/	G06F9/	G06F11/	G06F12/	G06F13/	G06F15/	G06K9/	G06K15/	H04L9/	Total
2003	1	0	1	0	0	0	0	2	0	2	6
2002	1	0	0	1	0	0	0	0	0	0	2
2001	1	0	0	0	0	0	0	1	1	1	4
2000	2	1	1	0	0	1	0	0	0	0	5
1999	1	0	0	0	0	0	1	1	0	0	3
1998	1	0	0	1	0	0	1	1	0	0	4
1997	0	0	1	1	1	0	3	1	0	0	7
1996	1	0	0	0	0	0	0	0	0	1	2
1995	0	0	0	0	0	0	0	0	0	0	0
1994	0	0	0	0	0	0	0	0	0	0	0
1993	0	0	0	0	0	0	0	0	0	0	0
Total	8	1	3	3	1	1	5	6	1	4	33

Note: no patents and patent applications in IPC number G06F5/ (left out in the table).
Source: http://dk.espacenet.com/.

[9] We do not have information other than publication number, publication date, inventor name, applicant name, priority number and IPC classification so that we cannot distinguish between a granted patent and a patent application.

[10] We cannot rule out, however, that there in fact are firms behind the patent applications by private persons, for example the firm owner.

Another issue here is that there are many foreign firms that hold or have applied for patents on software-related inventions. These are patents that come from the European Patent Office.

3. The Pros and Cons of Software Patents

Some economists argue that in the particular case of software-related patents, the negative aspects of patents by far overweigh the positive ones. Most of these arguments revolve around the new strategic importance of patents, and we will review the economic arguments for and against software patents below. To provide the reader an overview, we summarize what we see as the main arguments in bullet points and discuss them. We start with the pros and later outline the cons.

3.1 Arguments for Software Patents

The main argument in favor of software patents is the same as for any patent in general: it is supposed to enable the inventor to reap a fair share of the rents accruing from her efforts and thereby stimulates innovation and growth. On a similar note, let us point out that the distinction where software is patentable only if it is embedded in a machine is rather arbitrary. Ideally, the possibilities to patent an invention should depend on a number of factors such as the usefulness and novelty of the invention, the ease with which it can be imitated, and the cost of developing it – but not on the specific physical form that it takes.

Argument 1: *Software patents stimulate innovation in the software industry because they increase the return to R&D.*

The empirical evidence on the effects of patents (cited above) is in fact mixed and does not provide strong support for introducing software patents in Europe. Furthermore, the success of the patent system in stimulating R&D in some industries does not necessarily imply that the same holds true for the software industry. The special characteristics of innovation in the software industry such as short life cycles and sequential innovations could well tilt the balance against software patents. Unfortunately, the empirical evidence on this central question is very scarce. Bessen and Hunt (2003) suggest that software patents tend to decrease R&D in the software industry. However, an earlier version of the study by Bessen and Hunt has been criticized sharply by Hahn and Wallsten (2003) on technical grounds, so additional evidence is needed.

A patent is often a prerequisite for licensing a technology to another firm, because it establishes a property right that protects the licensor from opportunistic behavior on the side of the licensee, see e.g. Arora et al. (2001). Some observers, such as Niels Bo Theilgaard of the Danish software producer Navision argued in a public hearing that software patents will play an important role in creating a more efficient market for software technology.[11]

Argument 2: *Software patents will create a more efficient market for technology.*

Software patents are likely to lead to more licensing, but the difficult question is whether firms license more from each other because they have to (due to stronger intellectual property rights) or because there is a more efficient market for technology. Evidence, for example from questionnaires, might help to shed light on this question.

It has also been argued that software patents facilitate entry into niches of the software market. A patent provides the entrant with an exclusive right to its intellectual property that shields it from tough competition and facilitates outside financing.

Argument 3: *Patents support niche entry in IT markets*.

Until recently, US and Japanese firms had better possibility to patent their software inventions whereas European firms could, at least in principle, not. This has led some observers to suggest that European software firms were at a competitive disadvantage on the world market. Indeed, some members of the European Parliament such as the German representative Erika Mann pushed that argument even further by saying that introducing European software patents would save European workplaces since software inventions could then not be used by competing firms (Spiegel online, Sept. 1, 2003). Without software protection, so the argument goes, European ideas could be stolen by firms outside Europe so that jobs at the inventors' sites would be destroyed.

[11] Navision was later brought by Microsoft and Theilgaard now is general manager of Microsoft Business Solutions in Denmark.

Argument 4: *Software patents give European software firms a stronger position on the world market.*

Notice that this is not an argument for software patents as such, but an argument of a protectionist nature suggesting that Europe benefits from software patents when other countries have already introduced it.

3.2 Arguments against Software Patents

An argument often encountered is that seemingly trivial software inventions are granted patents with a very wide scope. One well-known example is the one-click-only online shopping patent granted to Amazon.com. This patent forces other internet shops to introduce unnecessary steps in online purchasing to make sure that only Amazon.com customers exclusively enjoy one-click shopping.

Argument 1: *Many software patents are of bad quality.*

This is to our mind not an argument against software patents in general, but an argument against the way that software patents have been implemented in the US. Software patents should, like all other inventions, have a novelty requirement that ensures that only useful and non-obvious inventions can be patented. There are several reasons why the novelty requirement apparently has been compromised for some software patents in the US. First of all, software is a new subject matter to the USPTO. Therefore, sufficient expertise and files of the prior art have been lacking. This has probably led to some mistakes by the USPTO, something that is also likely to happen in Europe. Furthermore, Cohen and Lemley (2001) argue that in the US, the disclosure requirement for software patents is much lower than for other types of inventions. Typically, software patentees do not have to reveal the implementing codes or even detailed descriptions of the invention. This effectively prevents other inventors from using the knowledge created and reduces the positive dynamic effects of patents outlined above. The lack of sufficient disclosure also gives rise to some practical problems, because it becomes difficult for the USPTO to specify the scope of a patent precisely and for other inventors to know whether they are infringing the patent. The European system should make sure that software inventions face the same disclosure requirements as other types of inventions.

Software patents may, as explained above, create a patent thicket. This increases development costs, because software firms would need legal advice, would need to search patent data bases, and might also need to license technologies from competitors. Survey evidence from Germany, for example, shows that 20 percent of the firms interviewed from the primary software sector and 40 percent of the firms from the secondary sector have been involved in patent trials (Blind et al. 2001). Another practical problem related to software inventions described by Jaffe (1999) is that property rights are hard to allocate both between and within firms and require lengthy and costly negotiations. Moreover, the administration of licenses imposes large cost.

Argument 2: *Software patents create a patent thicket that increases the cost of software development.*

Software patent critics say that a higher cost of software development due to software patents will ultimately lead to a slowdown in software innovations, which clearly is detrimental to the original goal of the patent system. David et al. (2003) cite a CEO of a SME who said at a Federal Trade Commission/Department of Justice hearing: "I have no idea whether my product infringes on an upward of 120 different patents, all of which are held by large companies who could sue me without thinking about it. The end result, much like Borland, I have now issued a directive that we reallocate roughly 20 to 35 per cent of our developers' resources and sign on two separate law firms to increase our patent portfolio to be able to engage in the patent spew conflict".

Another of the main arguments against software patents, which builds on the previous argument, is that SMEs are at a disadvantage relative to large enterprises when navigating the patent thicket.

Argument 3: *Software patents put entrants and other SMEs at a disadvantage.*

Software-related patents may, as argued above, increase the cost of software development. Such an increase might be prohibitively high for SMEs to sustain. On the contrary, large firms that run own law departments might easily afford the increased cost and might even receive returns on their legal investments through successful patent infringement trials. Given the fact that Europe has a relatively small commercial software industry, a US-like software patent legislation might put the domestic industry at a disadvantage. The argument is even stronger for Denmark that has a quite small software industry even by European standards as shown in Section 4 below. A last issue to be mentioned here is the idea of a patent trial insurance that was developed during the recent Danish EU presidency. The motivation here is to ensure that SMEs are able to survive patent infringement cases by establishing a market for patent insurances. SMEs would buy such an insurance and the insurer would pay for any outlays related to patent trials. Such insurances are indeed offered in the US. Due to the risky nature of the patent insurance business that is also characterized by severe problems of asymmetric information, insurance fees might, however, be prohibitively high; in particular for the group of firms they are geared to: SMEs.

A strong argument against patents to software-related inventions is about open source software, an issue that is particularly important for Europe due to the high importance of open source software (Bessen 2001; Gehring 2000; Horns 2000; Lutterbeck et al. 2000; Smets-Solanes 2000; Smarr and Graham 2000).

Argument 4: *Software patents will destroy the development of open source software.*

Many observers also take the open source community as a case against the patent system in general: apparently software can be developed without patents or even commercial interests. However, a closer look at the open source movement reveals that there are often economic incentives at work (which, however, does not in itself make a case for software-related patenting). The core contributors to an open source program are, for example, explicitly acknowledged, which serves as a positive signal about their abilities to the outside world. Lerner and Tirole (2002b) present an interesting economic analysis of open source software. Their study suggests that open source software does well for specialized software programs and for prestigious and ‘exciting’ software programs such as a new operating system but that less glamorous tasks such as the development of easy-to-use interfaces, hotline services, new drivers etc. are better provided by commercial firms. This view is challenged by real life evidence by authors such as Henkel (2003), Hertel et al. (2003), and Lakhani and von Hippel (2003). The future roles of commercial (closed source) and open source software do not seem to be the central issue here, however. After all, if the open source community would not be affected by the introduction of software patents, this should have no bearing on the decision whether to introduce software-related patents or not. More to the point, it has been argued that the non-commercial and decentralized open source movement lacks the ability to arm against patent infringement suits by commercial competitors, even though giants like IBM are getting more and more involved in open source. At the same time it also lacks the ability to sue commercial competitors for patent infringement. Microsoft for example used its patent portfolio against the open source fileserver software “Samba”, and it is to be expected that more cases against Linux and other open source software will follow (Henkel 2002a). The open source movement may thus have more difficulties surviving within a patent thicket than commercial firms. For this reason, authors such as Lutterbeck (2000) and Horns (2000) suggest to exclude open-source code from patentability to keep the open source community alive. Finally, it should be noted that a few authors, among them Nichols (1999), actually see improvements for the open source community by software patentability. According to his view, the codification of the invention and its publication in a patent leads to more competition and hence to more innovation.

In markets with strong network externalities, there is a tendency for dominant firms to become stronger over time, at least when products are incompatible. The market forces push towards standardization, and the dominant firms’ product often ends up as the *de facto* standard of the industry. Once such a standard is established, it is very hard to change. Customers switching to a new software product that is not compatible with the current standard would lose the benefit from being in a large network of customers using the same standard. Additionally, there are often investments, such as learning costs, that are specific to the standard and that are lost if an incompatible product is adopted. A new but incompatible product would therefore have to be drastically better than existing products to lure customers away. Microsoft’s operating system Windows is such an example. It has become standard and all products that are incompatible with Windows will not sell large volumes on the PC market. Coming back to patents, Farrell (1995) as well as Mazzeloni and Nelson (1998) have argued that in markets with strong network externalities, such as many markets for software, a patent will allow the patenting firm to establish its product as the industry standard. This proprietary standard is very hard to challenge even after the patent expires. A software patent may therefore create a monopoly position that lasts longer than in most other industries.

Argument 5: *Software patents and network externalities create a very strong monopoly position for the patent holder.*

The patent system has to balance dynamic and static efficiency in the market. Software patents may therefore be less beneficial to society than other patents, because the effective length of the monopoly granted by a patent is excessive relative to the cost of software development when there are strong network externalities.

3.3 The alternative to software patents: copyrights

Until recently software inventions were protected copyright only. Copyrights – legal terms that describe the rights given to creators for literary or artistic works – are used for the protection of source code (while patents are used for algorithms that is a part of the source code).[12] Copyright also comes automatically, at no cost, purely by writing the code. The main difference between patent and copyright protection is that patents protect ideas while copyrights protect their implementation only. It is thus much more easily available to open source developers than patents. An appealing feature of copyright protection is that those who develop new software can be sure that they do not violate others property rights. Copyrights do therefore not create thickets of overlapping property rights as patents risk to do. On the downside, software-related inventions that contain very novel and valuable ideas are likely to receive stronger protection from a patent as it prevents imitation by competitors. This suggests a natural division of labor between patents and copyrights where only the very novel inventions are protected by patent laws and all other inventions are protected by copyrights. This, in turn, requires that novelty requirement to obtain a software-related patent is set sufficiently high.

4. Implications for Denmark

4.1 Software production and consumption in Denmark

Hougaard Jensen et al. (2003) point out that Denmark plays a rather subordinate role in the *production* of Information and Communication Technologies (ICT), both with respect

[12] A study by Oz (1998) shows that copyright protection is used more often than patent protection even in the US where there are no bounds to software patenting.

to hardware and software. Denmark also lacks “big players” in the ICT industry like Nokia of Finland, Ericsson of Sweden or Bosch and Siemens of Germany.

Danish firms, Danish citizens and even the Danish public administration are, however, strong adopters of ICT. Hougaard Jensen et al. (2003) also show that the penetration of modern ICT technologies like e-commerce is particularly high in Denmark compared to other OECD countries.

Table 2, taken from van Ark (2002), displays the share of ICT-producing sectors in total GDP across selected OECD countries, where a distinction has been made between ICT services (e.g. software development, software implementation, programming services) and ICT products (e.g. hardware production). The size of the Danish ICT sector, whether measured by the share of ICT products or ICT services in total GDP, is relatively small. It is even below the average for the EU-15, and compared to leading ICT-producing countries such as the USA and Finland, the differences are substantial.

Table 2: *Share of ICT-producing sectors in total GDP (2000, in percent)*

	ICT products	ICT services
Denmark	1.1	3.6
Finland	5.6	4.5
Sweden	2.2	5.0
the Netherlands	1.3	5.1
Germany	1.5	3.9
UK	1.8	5.2
USA	2.3	5.3
EU-15	1.5	4.3

Source: Van Ark et al. (2002).

It is thus quite unsurprising that an empirical study by van Ark (2002) finds that the growth contribution of the ICT-*producing* sector has been particularly low for Denmark, whereas it has been particularly high for Finland. By contrast, however, the contribution from ICT-*using* sectors to GDP growth has been high for Denmark, especially with respect to labor productivity. This suggests that although Denmark is not an important ICT-producer, it still largely benefits from achievements made in information and communication technologies.

Figures provided by Statistics Denmark (2003), as shown in Table 2, also suggest that the Danish software sector has moderately grown between 1992 and 1999, both in terms of employment and in terms of the total number of firms. The software industry made up 2.9 percent of the total number of private firms and 1.2 percent of total full time employment in 1999. It is hence an important, but still relatively small industry.

There is a major flaw with Table 3 (and also with Table 4): Statistics Denmark uses the four-digit version of the NACE Rev.-1 classification of industries. Classification number 7220 ("Software consultancy and supply") comes closest to what one usually considers as software-related industries. Unfortunately, this definition apparently also consists of consultancy services so that the picture might be distorted. Many of these consultancy services are likely to be spent on customization of existing software so that distinguishing between pure programming and customization activities is not clear a priori.

Table 3: *Number of firms and number of employees in ICT industries in Denmark*

	Number of firms		Share in total			Number of employees		Share in		
	by sector		#" of firms (%)		Change	(full time equivalents)		total employment		Change
	1992	1999	1992	1999	(in %)	1992	1999	1992	1999	(in %)
IT manufacturing	1,010	898	0.4	0.4	0.0	20,756	18,819	2.2	1.7	-0.4
IT trade	3,240	2,639	1.4	1.2	-0.2	22,023	26,963	2.3	2.5	0.2
Telecommunications	32	151	0.0	0.1	0.1	13,094	18,489	1.4	1.7	0.3
IT-services	6,349	9,172	2.7	4.0	1.3	12,620	23,477	1.3	2.2	0.8
Development	5,681	6,763	2.5	2.9	0.5	7,590	13,551	0.8	1.2	0.5
Manufacturing	32,599	27,462	14.1	12.0	-2.1	396,079	413,404	41.2	38.0	-3.3
Services	160,083	162,292	69.1	70.7	1.7	448,715	533,765	46.7	49.0	2.3
Private sector	231,801	229,452	100	100	0	960,856	1,088,901	100	100	0
IT manufacturing	1,010	898	0.4	0.4	0.0	20,756	18,819	2.2	1.7	-0.4

Source: Statistics Denmark (2003).

Most of the software-producing firms in Demark are small enterprises. While in IT manufacturing, IT trade, telecommunications and IT-services in general larger firms with more than 100 employees dominate, 29 percent of the employees in the software sector work in firms with less than ten employees, 28 percent work in firms with between nine and 49 employees and ten percent work in firms with between 50 and 99 employees.

Table 4: *Size distribution of ICT firms 1998*

	Share of firms in the following size classes:					Total
	0-9	10-49	50-99	>100	# of firms	0-9
IT manufacturing	5.6	12.8	12.9	68.7	22,597	5.6

IT trade	12.9	23.5	11.3	52.3	31,278	12.9
Telecommunications	1.0	0.3	3.6	92.4	19,983	1.0
IT-services	23.9	21.5	10.4	44.2	29,767	23.9
Other software consultancy and supply	28.6	27.2	10.2	29.3	18,961	28.6
Manufacturing	11.4	19.7	10.2	58.9	497,203	11.4
Services	35.6	24.3	8.1	32.0	831,609	35.6
Private sector	28.6	23.7	8.6	39.1	1,523,504	28.6

Source: Statistics Denmark (2003)

4.2 Implications for Denmark

Even though Denmark is not a major producer of software, its stakes in the software patenting debate are high. Denmark is a major consumer of software and empirical studies have shown that even countries that do not produce their own software may benefit from IT-induced productivity growth. Van Ark et al. (2002) for example demonstrate that the productivity benefits from IT are of equal magnitude in Australia, a country with no significant software and hardware industry and the US. The Danish government would hence be ill advised not to take the software patenting matter seriously.

Denmark does not have a significant software industry, but the proposed patent legislation will be important for other industries as well. Under the new European legislation, embedded software, software that is integrated in products, for example in pumps, cellular phones and hearing aids can be protected. This type of patenting is less prone to block further developments than the patenting of “software as such”, so patents on embedded software inventions will probably be beneficial for Denmark.

Another tricky issue that should not be neglected is that there is a lot of (undesired) uncertainty around software patenting in Europe right now. In order to elaborate this problem in further detail, we interviewed Kurt Crone Jørgensen, the software representative of Grundfos A/S, the world-market leader in the manufacturing of pumps, who told us: “In one project we have recently decided against using open source software. There was a number of reasons behind this decision - one major reason was the (in my opinion) uncertainty regarding licenses. What are the exact demands/requirements/principles when using open source software - would we be forced to make parts of our software public?”

5. Policy conclusions and further research

5.1 Policy Recommendations

Proponents of software patents argue that they are an important device for inducing innovation in software as much as for other complementary technologies. Opponents of software-related patenting argue that it is often too easy to obtain patent protection in software and that firms use patents strategically, which creates “patent thickets” that increase the cost of innovation. Furthermore, it is argued that software patents will effectively put an end to open source software.

At this stage it is impossible to say who is right. There is simply not enough evidence on central issues such as the effects of patents on innovation in the software industry and the seriousness of the problems that patent thickets create. It is probably true, however, that software patents are less likely to be beneficial in Europe than in the US, because Europe, and even more so Denmark, do not host big players in the software industry. With few exceptions such as SAP, European countries are relatively unimportant producers of software – but adoption rates of ICT are tremendous. Moreover, in Europe a flourishing open source community exists that would be endangered if software patents existed. As Bruce Perens, one of the most important driving forces behind the open source community, recently put it in a conference statement: “If you want to kill open source software – software-related patents are the way to do it.” However, if it is decided to allow software patents – in spite of the lack of conclusive evidence – the opponents’ arguments do suggest a number of concerns that should be taken seriously when implementing software patents in Europe. The European system should ensure that software inventions face the same disclosure requirements as other types of inventions. Disclosure is a prerequisite for a transparent and well-functioning patent system where: (i) the patent office can evaluate the size of inventions properly to determine when novelty requirement is satisfied; (ii) an inventor receives a patent on a precisely specified claim; (iii) the public can study the patent application to learn about the invention; (iv) other firms can challenge patents that contain too broad claims. Cohen and Lemley (2001) also argue that limited reverse engineering of software programs that contain patented elements should be allowed. This is another way to make sure that sufficient knowledge about the invention is disclosed to the public.

The main argument against software patents is that they create hold-up problems due to the sequential and complementary nature of innovations in the software industry, which may result in a patent thicket. At the more general level we will argue that some degree of hold-up (or market power) is not problematic. Indeed, research on patenting of sequential innovations suggests that an inventor should receive some compensation when a later inventor uses her invention as an input, and a licensing fee ensures precisely this (Scotchmer 1991 and Green and Scotchmer 1996). Hold-up is a problem only if the inventor is able to extract an excessive amount of rents from later inventors relative to the usefulness of the invention. DKPTO and other patent authorities in Europe will therefore play a crucial role in balancing the incentives of different inventors, a difficult problem to tackle. The practical problems concerning the optimal length and scope of software patents deserve careful analysis, but the above analysis does lead to a few preliminary conclusions. The risk of hold-up suggests that European patent authorities should be very cautious towards granting patents with very broad claims. This is especially true in the beginning where it is very hard to find out how novel an invention is. Second, it may be necessary to require licensing of essential patents (e.g. patents on the central parts of a standard) at fair prices. This should be feasible, as patent laws already contain provisions allowing for mandatory licensing of essential facilities. Although the idea suffers from the severe drawback that it might be prohibitively difficult to first determine which software-related invention is important and then to set the appropriate licensing fee. Finally, it has been discussed to introduce a "grace period" for inventions that enable the early publication of inventions without ruling out later patentability.[13] This would alleviate the problem that inventions have to be kept secret during the often lengthy patent application process, a serious problem in the fast-moving software industry. Gehring (2000) and Horns (2000) discuss this idea in further detail.

5.2 Future Research

We have already pointed out at various instances in this paper that researchers and politicians alike know very little about the likely consequences of changes in the patentability of software-related inventions. There are many questions that are left unanswered by the literature we reviewed in this paper, and we have already pointed out

[13] Germany for example used to allow for an early publication that did not rule out later patent applications until the end of the 70s. The grace period was six month and was abolished in the context of the European patent treaties of 1978/1979.

research areas that we find particularly promising in the context of software patenting above. In the US, software patents have now been issued for more than a decade. While there are a few empirical studies trying measure the effect of patents on innovation in the US software industry, there clearly is work missing on this important topic. Another central issue is the existence of patent thickets and the problems that they create. Here, more theoretical work is needed to provide a deeper understanding of how firms navigate the thicket and what public policy should look like. Additional empirical evidence on patent thickets, for instance from questionnaires or case studies, would also be valuable.

References

Arora, A., Fosfuri, A., and A. Gambardella (2001). Markets for Technology. The Economics of Innovation and Corporate Strategy. MIT Press.

Arundel, A. and I. Kabla (1998). What percentage of innovations are patented? Empirical estimates for European firms, Research Policy 27, 127-141.

Bessen, J. (2001). Open source software: free provision of complex public goods, http://www.researchoninnovation.org/opensrc.pdf.

Bessen, J. and R.M. Hunt (2003). An empirical look at software patents, MIT working paper.

Blind, K. (1999). The impact of intellectual property rights on the propensity to standardize at standardization organizations: an international cross-section analysis, paper presented at the International telecommunications Society, Turin.

Blind, K., J. Edler, R. Nack and J. Strauss (2001). Mikro- und makroökonomische Implikationen der Patentierbarkeit von Softwareinnovationen: geistige Eigentumsrechte in der Informationstechnologie im Spannungsfeld von Wettbewerb und Innovation, report commissioned by the German Ministry for Economics and Technologie, Karlsruhe.

Cohen, J.E. and M.A. Lemley (2001). Patent Scope and Innovation in the Software Industry. California Law Review, 89, 1-58. Available at www.SSRN.com.

Cohen, W.M., P.P. Walsh and R.R. Nelson (2000). Protecting their intellectual assets: appropriability conditions and why firms patent or not?, NBER working paper 7552.

Computerworld (2004). Microsoft ordered to pay Eolas $520M in patent case; http://www.computerworld.com/developmenttopics/websitemgmt/story/0,10801,89086,00.html

David, P.A., D. Foray, B.H. Hall, B. Kahin and W.E: Steinmüller (2003). Should Europe encourage more patenting of software? A research-based primer for policy makers; www.researchineurope.org/draft/review.htm.

European Commission (2000). The patentability of computer-implemented inventions, Directorate General Internal Markets, Brussels.

Farrell, J. E. (1995): “Some Arguments for Weaker Intellectual Property Protection in Network Industries,” in Standards Policy and Information Infrastructure, edited by B. Kahin and J. Abbate, MIT Press, 368-377.

Gehring, R. (2000). Berliner Ansatz zu Open Source software patents, http://130.149.19.71:8080/Think-Ahead.ORG/Cyberlaw.
Graham, S.J.H. and D.C. Mowery (2002b). Intellectual property protection in the US software industry, in: Cohen, W. and S. Merril (Eds) The patent system in the knowledge-based economy, National Academies Press, Washington.
Graham, S.J.H. and D.C. Mowery (2002b). Submarines in software? Continuations in U.S. software patenting in the 1980s and 1990s, University of California at Berkeley working paper.
Griliches, Z. (1984). R&D, patents and productivity, University of Chicago Press, Chicago.
Griliches, Z., A. Pakes and B.H. Hall (1987). The value of patents as indicators of innovative activity, in Dasgupta, P. and P. Stoneman (Eds): Economic Policy and technological performance, Cambridge University Press, Cambridge.
Hahn, R.W. and S. Wallsten (2003). A review of Bessen and Hunt's analysis of software patents, AEI-Brooking Center for Regulatory Studies mimeo.
Hall, B.H. (1996). The private and social returns to research and development: what have we learned?, in Smith, B.L.R. and C.E. Barfield (Eds): Technology, R&D and the economy, The Brookings Institution and the American Enterprise Institute, Washington, D.C.
Hall, B.H. and R.H. Ziedonis (2001). The determinants of patenting in the US semiconductor industry, 1980-1994. Rand Journal of Economics 32, 101-128.
Henkel, J. (2002). Zuviel Schutz schadet nur, VDI Nachrichten, Nov. 11, 2002.
Henkel, J (2003). Software development in embedded Linux - informal collaboration of competing firms, forthcoming in: proceedings 6. Internationalen Tagung Wirtschaftsinformatik 2003, Dresden.
Hertel, G., S. Niedner. and S. Herrmann (2003). Motivation of software developers in open source projects: An internet-based survey of contributors to the Linux kernel. Research Policy, 32, 1159-1177.
Horns, A.H. (2000). Der patentschutz für softwarebezogene Erfindungen im Verhältnis zur "open source"-Software, JurPC internet document.
Jaffe, A.B. (1999). The U.S. patent system in transition: policy innovation and the innovation process, NBER working paper 7280.
Hougaard Jensen. S.E., N. Malchow-Møller, J. Rose Skaksen, A. Sørensen (2003). Denmark and the Information Society: Challenges for Research and Education Policy, Copenhagen: DJØF Publishing.
Kash, D.E. and W. Kingston (2001). Patents in a world of complex technologies, Science and Public Policy, 28(1), 11-22.
Lakhani, K.R and E. von Hippel (2003). "How Open Source Software Works: Free User to User Assistance." Research Policy 32(6): 923-943.
Works: Free User to User Assistance." Research Policy 32(6): 923-943. ... Lerner, J. (2002a): "150 Years of Patent Protection," *American Economic Review Papers and Proceedings*, 92, 221-225.
Lerner, J. (2002b): "Patent Policy and Innovation Over 150 Years," NBER WP No. 8977.
Lerner, J. and J. Tirole (2002a). Efficient Patent Pools. Mimeo. Available at: http://www.SSRN.com.
Lerner, J. and J. Tirole (2002b). Some Simple Economics of Open Source. *Journal of Industrial Economics*, L(2), 197-234.

Lutterbeck, B., R. Gehring, A.H. Horns (2000). Sicherheit in der Informationstechnologie und Patentschutz für Softwareprodukte – ein Widerspruch? Research report prepared for the German Federal Ministry for technology and the Economy, Berlin.

Mazelloni, R. and R. Nelson (1998). The benefits and cost of strong patent protection: a contribution to the current debate, Research Policy 72, 273-284.

Murillo, G. (1998). Institutional development in the software industry: intellectual property protection. UMI dissertation abstract, Ann Arbor.

Nalley, E.T. (2000). Intellectual property in computer programs, Business Horizons 43(4), 43-51.

Nichols, K. (1999). The age of software patents, Computer 32(4), 25-31.

PbT Consultants (2001). The results of the European Commission consultation exercise on the patentability of computer implemented inventions, mimeo.

Shapiro, C. (2001). Navigating the Patent Thicket: Cross Licenses, Patent Pools and Standard Setting. *Innovation Policy and the Economy*, Adam Jaffe, Joshua Lerner, and Scott Stern, eds., National Bureau of Economics, 2001.

Smarr, L. and S. Graham (2000). Recommendation of the panel on open source software for high end computing: president's information technology advisory committee, mimeo.

Smets-Solanes, J.-P. (2000). Software useright: solving inconsistencies of software patents, paper presented at the second Nordic European/USENIX conference, Malmø.

Smets-Solanes, J.-P. (2001). Stimulating competition and innovation in the information society, working paper downloadable at http://www.pro-innovation.org.

Spiegel online (2003). Software-Patente?; http://www.spiegel.de/spiegel/0,1518,263719,00.html.

UK Patent Office (2001). Software patenting consultation; http://www.patent.gov.org.uk/news.softpat.htm.

Van Ark, B., R. Inklaar and R.H. McGuckin (2002). Changing gears – productivity, ICT and service industries: Europe and the United States, paper presented at the 2. ZEW conference "Economics of Information and New Technologies", June 24-25, Mannheim.

Appendix: Danish software patents and patent applications in detail

Title	Application/ number	Applicant	Publication date
Danish patent applicants/holders			
Private			
Globe-PC	DK200301609	Olav Benndbæk (DK)	2003-10-30
Apparat samt fremgangsmåde til fritlægning af dele af digitalt repræsenterede billeder	DK88594	OLSEN MORTEN (DK); TOKSVIG MICHAEL JOHN (DK)	1996-01-29
Fremgangsmåde til brugerindstilling samt til bevægelseskorrektion af en inputanordning, et lagermedium samt en inputanordning	DK148597	KANITECH A S (DK)	1999-06-19
Fremgangsmåde og apparat til hygiejnisk beskyttelse af berøringsflader til betjening, registrering og/eller fremfinding af (…)	DK27599	BUCHHORN JOHAN (DK)	2000-09-02
Håndledsstøtte med indbygget pegeredskab	DK19797	KRISTENSEN KURT (DK)	1998-08-25
Sum-intervaldetektor	DK174398	MOELLER OLE (DK)	2000-06-30
Fremgangsmåde til forhindring af uautoriseret brug af EDB-program	DK2997	JESSEN HANS (DK)	1997-01-14
Metode til objektivisering af subjektive klassifikationer	DK128796	MEYROWITSCH JAN (DK)	1998-05-15
Corporate			
Fremgangsmåde til fremvisning og behandling af information fra internettet	DK200101775	SENIEURO APS (DK)	2003-05-31
Fremgangsmåde samt styre- og overvågningssystem til detektering og indikering af et systems tilstande	DK69796	DSC COMMUNICATIONS AS (DK)	1997-12-25
Fremgangsmåde og anlæg til brug ved behandling af et kødemne	DK167462	SLAGTERIERNES FORSKNINGSINST (DK)	1999-11-01
Databehandlingssystem og fremgangsmåde til beregning af størrelsen af finansielle instrumenter	DK16596	REALKREDIT DANMARK A S (DK)	1997-08-03
Fremgangsmåde og databehandlingssystem til bestemmelse af finansielle instrumenter til anvendelse ved finansieringen af et lån, der er i det mindste delvist refinansieret under dets løbetid	DK12597	REALKREDIT DANMARK A S (DK)	1997-08-03
Fremgangsmåde og apparat til styring af lysintensitet i forbindelse med eksponering af fotofølsomt materiale	DK200000614	PURUP ESKOFOT AS (DK)	2001-12-12
System til elektronisk udlevering af en personlig identifikationskode	DK174672	ORANGE AS (DK)	2003-08-25

Foreign patent applicants/holders			
System til at foretage automatisk backup af filer, når en personlig computer slukkes	DK200200265	SERICHOL BLASCO JOSE MARIA (ES)	2002-08-21
Styrepanel i form af en berøringsfølsom skærm med styring ved glidende berøring, samt en fremgangsmåde til betjening af anordningen	DK173622	SPERRY MARINE INC (US)	2001-05-07
Regulator til styring af en køleenhed	DK173285	MITSUBISHI HEAVY IND LTD (JP)	2000-06-13
Databehandlingsapparat	DK171728	AMDAHL CORP (US)	1997-04-14
Fremgangsmåde og indretning til frembringelse af et startsignal for parallel-synkron drift af tre identiske databehandlings (…)	DK172489	ERICSSON TELEFON AB L M (SE)	1998-10-05
Kommunikationsdatamat til et pakke-switchet netværk	DK173266	SPRINT INTERNAT COMM CORP (US)	2000-05-29
Fjerndatasystem, især til anvendelse i forbindelse med en salgsautomat	DK171875	PEPSICO INC (US)	1997-07-21
Bærbar enhed omfattende biometrisk baserede verificeringsegenskaber	DK200200630	TREK 2000 INTERNAT LTD (SG)	2003-01-09
Databærer med en optisk ægthedsafprøvning samt fremgangsmåde til fremstilling og afprøvning af databæreren	DK173812	GAO GES AUTOMATION ORG (DE)	2001-11-12
Apparat til behandling af dokumenter	DK173066	BANCTEC INC (US)	1999-12-13
Billedbehandlingsapparat samt et video-kodningsapparat, der anvender et sådant billedbehandlingsapparat.	DK171882	BRITISH TELECOMM (GB)	1997-07-28
Nøglestrømsgenerator med tilbagekoblet skifteregisterstruktur	DK174597	GEN INSTRUMENT CORP (US)	2003-07-14
Fremgangsmåde til afsendelse af hemmelige nøgler til sikkerhedsmoduler og brugerkort i et databehandlingsnet, samt anvendelse af fremgangsmåden i et databehandlingsnet	DK171320	BULL CP8 (FR)	1996-09-02