13 Tex. Intell. Prop. L.J. 171

Texas Intellectual Property Law Journal
Winter 2005

Articles

THE CONTINUING DEBATE OF SOFTWARE PATENTS AND THE OPEN SOURCE MOVEMENT

Grant C. Yang[a1]

Table of Contents

*172 Introduction

The United States, the European Union, and the international community have tried to harmonize patent law for decades.[1] Although several treaties have been proposed to harmonize some procedural aspects of patent law, patentable subject matters continue to remain a major point of contention and a major impediment to harmonization.[2] The United States has been on the forefront of broadening the range of patentable subject matter[3] because intellectual property is becoming the most important asset of the U.S. economy.[4] Although foreign patent offices have been wary of following the lead of the United States Patent & Trademark Office (USPTO), foreign patent offices frequently keep the overarching goals of harmonization in mind when establishing patent policy.[5]

***173** A Proposal for a Directive on the Patentability of Computer-Implemented Inventions (hereinafter Proposed Directive)[6] was working its way through the European Parliament. It had passed the European Parliament's first reading, the Competitive Council had agreed on a "common position," and the Parliament was preparing for the second reading of the directive.[7] All indications thus far had indicated that the European Union was moving towards harmonization with the United States. However, Members of the European Parliament (MEPs) from 13 countries had motioned to restart the legislative process on the Proposed Directive, citing that many recently elected MEPs "did not take part in the initial discussion on the directive."[8] Finally, the Legal Affairs Committee of the European Parliament (JURI)[9] voted to have the Commission restart the legislative process, requiring a new first reading guided by a new rapporteur, Michel Rocard.[10] The JURI request was ratified by the European Parliament, approved without debate by the Conference of President, and passed to the European Commission.[11] The European Commission "declined the European Parliament's request for a restart of the legislative process."[12] Over protests from several EU Member States and the open source community, ***174** the EU Council adopted the software patent directive.[13] Currently, after the EU Council's adoption of its Common Position on the Directive, the European Parliament has "three months to reject or amend the proposal," requiring a majority vote for every rejection or amendment.[14]

The United States has strongly supported software patents and had even previously threatened to leave harmonization negotiations regarding the single patent law (SPLT) if subject matter patentability issues are not met.[15] Although business methods would seem a likely subject matter that follows as a result of this Proposed Directive,[16] this paper will focus primarily on the arguments directed to the harmonization of software patent law and whether or not the European Union should have software patents, rather than argue for specific provisions or wording of an EU Directive.

Historically, the argument against computer software patents was that programs were algorithms or mathematical formulas and not patentable subject matter, thus not even discussing the real issue of whether software patents promote innovation.[17] After the U.S. courts shifted their viewpoint, the primary arguments, particularly those currently voiced by open source supporters contesting the EU Directive, are that software patents do not promote but rather inhibit innovation.[18] Opponents of software patents argue that copyright protection is enough for software ***175** and that innovation is plentiful in the software industry, citing the open source movement as an example.[19]

This paper will argue the case for computer software patents by surveying the current landscape of computer software protection in the European Union and the United States and discussing the harmonization efforts of the European Union

through its new directive. This paper will also describe characteristics of the software industry and the open source movement. Given the nature of the software industry, this paper will argue that software patents fulfill the goals of patent policy of promoting innovation. Also, due to the potential instability of an open source regime, rather than abolish software patents, this paper argues that software patents and open source should continue to work together harmoniously to promote innovation in the industry, as the case has been for the past decade.

I. The Current State of Law in Intellectual Property Protection for Computer Software

A. United States Approach

The computer industry is one of the largest industries in the United States,[20] and lawsuits concerning software patents are fought with increasingly high stakes.[21] The United States primarily employs two methods of legal protection: copyright and patents.[22]

1. Copyright Protection

Even before the Copyright Act was amended in 1976, the Copyright Office accepted computer programs as literary works.[23] The National Commission on ***176** New Technological Uses of Copyrighted Works (CONTU),[24] which Congress had established, concluded that copyright should protect software, but it did not necessarily preclude a patent regime.[25] Copyright law only protects expression, thus the literal code of a computer program is protectable but the underlying ideas are not.[26]

Several copyright cases have tried to define the boundaries of copyright protection while struggling with the idea-expression dichotomy embodied by 17 U.S.C. § 102(b).[27] In Apple Computer, Inc. v. Franklin Computer Corp., the Third Circuit extended copyright protection from the source code to the compiled object code, thus protecting the code regardless of the medium in which it is fixed.[28] In Whelan Associates v. Jaslow Dental Laboratory, the Third Circuit protected the essence of the computer program's functionality by recognizing non-literal elements of a computer program.[29] It held that copyright protection of computer software extended beyond literal code to a program's "structure, sequence, and organization."[30]

More recently, the Second Circuit in Computer Associates International, Inc. v. Altai, Inc. criticized the Whelan court for its mischaracterization of software embodying an idea as a whole rather than a program consisting of many subroutines, each consisting of an "idea."[31] The Altai court then put forth its own rule of "abstraction-filtration-comparison."[32] Most circuits have subsequently adopted the Altai ***177** formulation, which academics contend has severely limited the availability of copyright protection for computer software.[33] The First Circuit in Lotus Development Corp. v. Borland International, Inc. concluded that the software user interface, in particular the command menu hierarchy of the Lotus 1-2-3 spreadsheet program, was a "method of operation" and not copyrightable subject matter.[34] The holdings of Altai and Lotus narrowly allow copyright to "protect little more than the exact copying of software code."[35] However, despite its limitation on copyright protection of computer software, the Altai court made these decisions in light of the awareness of patents as an alternative protection regime.[36] They also noted that copyright is "not ideally suited to deal with . . . computer science."[37] Ultimately, courts have narrowed the availability of copyright to protect much of anything past the pure expression of the idea. However, they legitimized the need for a form of protection for computer software that exceeded the mere literal elements to protect the functions and ideas.

2. Patent Protection

Like in copyright law, the acceptance of computer software as an appropriate subject matter was gradual in patent law. U.S. courts had long been torn regarding the expression and idea aspects of software.[38] Even in Apple Computer, the court ***178** deflected an attack on operating systems as "methods" inappropriate for copyright protection by stating that Apple was not seeking to protect the method but rather the copyrighted source code "instructions."[39] The court then acknowledged that the functionality of the program, if it were to be protected, would be under patent law, which at the time was an unresolved issue under prior Supreme Court holdings.[40]

Under Diamond v. Diehr,[41] the Supreme Court "expressly held for the first time that . . . an invention was not necessarily unpatentable simply because it utilized software."[42] This decision went against a line of cases that had denied patents for computer software.[43] The Diehr patent application claimed a process for molding synthetic rubber using a computer to

constantly measure the temperature inside the mold so as to recalculate the cure time by means of the Arrhenius equation.[44] Diehr was distinguished from its previous holdings because those cases held that "an algorithm, or mathematical formula, is like a law of nature, which cannot be the subject of a patent," whereas the Diehr application was for a process of curing rubber which incorporated the use of a computer.[45]

Other cases solidified the position that computer software was patentable by further distinguishing mathematical equations from patentable subject matter. In In re Alappat,[46] the court found the applicant's invention to be patentable.[47] Alappat's invention used anti-aliasing techniques to create a smooth waveform display in a digital oscilloscope.[48] The Federal Circuit stated that while many elements of the ***179** invention performed mathematical functions, the claimed invention as a whole is directed to a combination of interrelated elements which combine to form a machine for converting discrete waveform data samples into anti-aliased pixel illumination intensity data to be displayed on a display means. This is not a disembodied mathematical concept which may be characterized as an "abstract idea," but rather a specific machine to produce a useful, concrete, and tangible result.[49]

The court stated that computer software was patentable so long as it meets the statutory patent law requirements of Title 35.[50] The concession by the courts to grant patents for a computer program in conjunction with a process indicated the Federal Circuit's full-fledged allowance of computer applications.[51] Despite the belief that the State Street Bank[52] holding sets a precedent that acceptance of software patents would inevitably result in allowing business method patents, the European Union does not support this trend.[53] Also, the arguments in this paper will focus solely on software patents.

***180 B. Intellectual Property Law of the European Community[54]**

Differences in national legal protection have negative and disruptive effects on innovation in particular technological growth areas for the European Community.[55] For this reason, the European Community has an increasing drive to become involved in the harmonization of intellectual property laws. The European Union is in a unique position regarding patent law because its internal harmonization efforts affect many countries.[56] In 1978, the European Patent Convention (EPC) substantially harmonized patent prosecution within the European Union and established the European Patent Office (EPO)[57] as a central office to receive and examine patent applications.[58] The driving force behind the EPC was a "pragmatic response to the need for industry and commerce to reduce the cost of securing patent protection across national boundaries within Europe."[59]

***181** Unlike the USPTO, the EPO is an agency that solely handles patent applications.[60] All applications that are made to a national office are eventually transmitted to the EPO for review, but infringement and revocation proceedings are left to the national courts.[61] Although an EPO patent attributes the same bundle of rights as that of a patent granted by the national office of a Member State,[62] the individual nations can still confer their own separate patent rights.[63] Hence, a patent granted by the EPO provides a patentee with a means to initiate an action; however, the validity and value of the patent is determined by national courts. While EPO Board decisions provide persuasive authority, national courts are bound by their own legal precedents, even if they create dissonance in EPC interpretation.[64] This raises concerns that national courts who do not uniformly interpret substantive patent law matters will negate the EPO's efforts of harmonization and efficiency.[65]

The goal of the EPC is to provide a means for efficiently granting the bundle of rights associated with patents.[66] This necessitates significant harmonization of the largely disjointed approaches to substantive patent law, particularly in the area of patentability.[67] The EPC specifically delineates subject matter that shall not be regarded as inventions, in particular, "rules and methods for performing mental acts, playing games or doing business, and programs for computers,"[68] and these provisions are excluded to the extent that a European patent relates to such subject matter "as such."[69] While the EPO has expressed its viewpoints on this matter, there is indication that Member States are leaning towards diverging interpretations. The European Parliament has chosen to rectify the confusion with a Proposed Directive on the Patentability of Computer-Implemented Inventions, discussed later in ***182** the paper.[70] While the EPO has already granted 30,000 computer program or business method-related inventions since 1986,[71] the passage of the Proposed Directive will not only maintain the status quo but will also strengthen the acceptance of patents for computer software patents in national courts.

1. Copyright Protection

The European Community had chosen copyright as the main method of protection of computer programs through the Council Directive on the Legal Protection of Computer Programs.[72] Computer programs are protected as literary works within the meaning of the Berne Convention.[73] Copyright protects a "computer program" expressed "in any form," including

"preparatory design work leading to the development" of the computer program.[74] The Directive makes it clear that it shall apply only to the "expression" of a computer program, as is typical of copyright law, and not to "[i]deas and principles which underlie any element of a computer program, including those which underlie its interfaces."[75]

2. Patent Protection

EPO cases have clarified the exclusion of subject matter regarding computer programs. In VICOM/Computer-related invention[76] the Board of Appeals considered the patentability of a "method of digitally processing images" using certain mathematical algorithms[77] "which can be carried out on a conventional general ***183** purpose computer."[78] Under EPC art. 52(2)(c) and (3), a computer program is excluded as subject matter "as such," but this was interpreted by the Board not to exclude claims directed to a technical process carried out by a computer program.[79] The Board of Appeals declared that the novel technical feature[80] existed in the claims because the invention "confers a technical benefit, namely a substantial increase in processing speed compared with the prior art."[81] The Board stated that the process of digital filtering consisted of "physical manipulation of electrical signals."[82] The Board also stated that "even if the idea underlying an invention may be considered to reside in a mathematical method a claim directed to a technical process in which the method is used does not seek protection for the mathematical method"; rather, if the claims outline the "technical means for carrying out the functions," then the conditions for declaring the "technical features" of the invention are met.[83] In other words, if a computer is used in conjunction with or applied to a process within the subject matter of patentability, then the claimed invention is patentable.[84]

These holdings have been reiterated in other cases since VICOM. The Technical Board of Appeals in IBM/Computer Programs considered the patentability of claims directed towards: (1) a computer program directly loadable into internal memory; and (2) a computer program stored on a computer usable medium.[85] The Board again gave a narrow reading of EPC art. 52(2), stating that the "legislators ***184** did not want to exclude from patentability all programs for computers . . . . [T]he fact that only patent applications relating to programs for computers as such are excluded from patentability means that patentability may be allowed for patent applications relating to programs for computers where the latter are not considered to be programs for computers as such."[86] Computer programs must have a technical character to be patentable.[87] Technical character was defined by the Board as requiring a "technical effect" which is "achieved by the internal functioning of a computer . . . under the influence of [a] . . . program."[88] The current EPO Guidelines for Examination reflect the EPO holdings and state that while computer programs are excluded as such from EPC art. 52, software is not excluded subject matter if it has a "technical character" by bringing about a "further technical effect."[89] Despite these holdings from the Board and the Guidelines of the EPO, "the existence of different legal traditions" have caused some divergences in the interpretation of the development of software patent law and necessitated a directive to harmonize the laws.[90]

3. Proposed Directive on the Patentability of Computer-Implemented Inventions

The most recent development in Europe which has caused the issue of software patents to flare up on the international realm was the proposal for a Directive on the patentability of computer-implemented inventions (hereinafter "Proposed Directive").[91] Article 2 defines "computer-implemented invention" as an invention realized "wholly or partly" by means of computer programs.[92] Article 4 reaffirms many of the EPC statutory requirements and EPO Board's holdings which require ***185** that software patents contain an inventive step and make a technical contribution.[93] Article 5 allows product claims in conformance with Article 27(1) of the Trade Related Aspects of Intellectual Property Rights (TRIPS) agreement.[94] Article 6 ensures that decompilation and interoperability provisions of Directive 91/250 EC are not affected by patent protection.[95] The Proposed Directive does not otherwise affect the normal patentability requirements in the EPC. Labour MEP Arlene McCarthy, the Rapporteur for the Proposed Directive, asserts that without the Directive there will be further confusion and uncertainty regarding the availability of software patents.[96] Though the legislative process for the Directive is on course, opposition is still strong and the ultimate result of the Directive is unclear. However, as this paper will argue the Directive should ultimately be approved as software patents can harmonize the international patent doctrine necessary to create a standard for national courts to abide by.

II. The Computer Software Industry and the Open Source Movement

A. What is the Software Industry?[97]

The software industry traditionally consists of companies that make applications, Internet services, and so forth. However,

the number of companies that would actually be affected by a potential ban on software patents is a greater number because today software is an integral component to most electrical devices. In these cases, the value of software is in determining its ability to enhance the sale of the core product. For example, if a consumer buys a car, software is an integral component, but the company sells and advertises its cars, not its software. On the other hand, the open source industry does not sell software and is therefore unaffected by a software patent ban, as will be explained in the characterization of the open source movement below. The traditional software industry has several key characteristics that may be taken into account when determining the best form of intellectual property protection.[98]

***186** 1. Market Position Created by Standards or "Network effects"

Due to the complex nature of computer software, successful companies typically have a large user base that is accustomed to interfacing with its products. Market dominance in the software industry is a result of users becoming accustomed to a certain standard[99] or a large population of users in an enclosed system that entices users to join the network.[100] For example, many users may prefer Microsoft's Office suite over Corel's WordPerfect suite because they are familiar with the product and also because Microsoft has a significant market share in this area.[101] Many users are accustomed to working in a certain file format or software interface. An example of the impact of network effects are large social networks such as Friendster.com, online auction sites such as Ebay.com, or instant messaging programs such as AIM, which proliferate because users are "locked in" due to the overwhelming de facto standardization of these programs.[102]

Although interfaces and standards are particularly easy to replicate,[103] a lead time can provide the necessary network effect to gain market share.[104] New users, if choosing between two similar interfaces, will choose the program that has a larger network of users.[105] In theory, a company that enters a market first is able to build a large user base, such that when a new program is introduced into the market, the first company is already entrenched in the market. Furthermore, once users are accustomed to one version of software, they are more likely to buy the next generation of software because they already understand the product's features and ***187** interface.[106] Therefore, patents on popular standards, which are needed for interoperability, can provide broad monopoly power and a strong bargaining position.[107]

The worry of some economists is that providing one supplier with a monopolistic lead produces a "tipping" effect where the lead is cumulative and difficult to counteract, leading to antitrust concerns.[108] On the other hand, due to the nature of the industry, there are various factors to counteract a tipping effect to create a level playing field. The computer software field has developed a "patent thicket," a "dense web of overlapping intellectual property rights that a company must hack its way through in order to actually commercialize new technology."[109] A firm with a software patent will not be able to gain a complete monopoly because it will also have to deal with cross-licenses, patent pools, and standard setting issues in its technologies that overlap with patent portfolios of other firms.[110]

2. Innovation is Rapid and Incremental

The software industry evolves through frequent but incremental innovations.[111] Change is so rapid that "entire product life cycles sometimes pass before patents can be issued."[112] However, it can be argued that this characteristic is no different than many other industries.[113] The confusion or appearance that software has comparatively rapid changes is largely due to the lack of breakthrough technologies versus incremental improvements of existing technologies.[114] Critics of software patents argue that "software innovation comes from programmers solving ***188** problems while developing software, not from projects whose specific purpose is to make inventions and obtain patents."[115] Because of this "research" process, programmers "throw away more 'inventions' each week than other people develop in a year."[116] Though innovations are copied and imitated quickly, this may in fact be a reason to have software patents as opposed to not having them. Startups or smaller companies with fewer resources would find it necessary to have patent protection in order to enter a market filled with larger competitors. If network effects have a strong impact, then patents would be necessary to cordon off a particularly innovative technology which would give another company time to create market branding and woo customers over to its own product.

B. What Is "Computer Software" ?

Software is generally defined as "a set of statements or instructions which is capable of causing a machine, having information processing capabilities (a computer), to perform a set of functions to achieve a result."[117] Software frequently consists of the "automation of known tasks,"[118] for example, word processing as an extension for paper. However, the automation of a known task should not be a prima facie indication of a lack of technical contribution to the art because the

implementation and paradigm required for automation are often innovations that can be worthy of protection.

The value of programs is in their compilations of various algorithms to produce a useful behavior.[119] Programs are advertised for their features and not for the textual source code that provides those features.[120] A computer program has an overarching functionality; however, the building blocks that programmers use to build a program to achieve functionality consist of modular subroutines.[121] In many ***189** industries "innovation lies in finding parts that construct a more valuable whole. While originality has merit, the true value lies in packaging. How the individual pieces are assembled into a whole is more important than the pieces themselves."[122] Most software products have a core functionality that is well-known, but the selling points are the features, possibly novel and patented ideas, that are added to each new version.

Although the costs to develop a computer program are low, the bulk of the value of a program is not in the literal elements of the code but rather the underlying ideas.[123] However, due to the nature of software, the research and programming are often intertwined, and therefore costs of research may often be attributed to costs of programming and vice versa. Although there may be less fixed costs in software development, such as manufacturing and distribution, competitiveness in the software industry largely depends on intellectual capital, such as software designers, programmers, and programming man-hours. Courts recognize that the "expense and difficulty in creating computer programs is attributable to the development of the structure and logic of the program, and to debugging, documentation and maintenance."[124] Intellectual capital can be costly and is a determinant factor in the speed to enter a market and to remain dominant by continuously updating software with new features.[125]

C. The Open Source Movement

The open source movement consists of two groups working together:[126] (1) the programmers who write parts of the software, and (2) the companies that service the software, that provide the commercial component to the open source movement.[127] The open source industry's marketable aspect is the service and support and not the actual open source software. Professor Benkler calls open source ***190** the "commons-based peer production," a growing movement where software is licensed so that any programmer can modify the source code freely, so long as the modifications are kept in the public domain.[128] The most ubiquitous open source license was that created by Richard Stallman[129] of the Free Software Foundation (FSF), who believed that patents were damaging to the industry.[130] He created the GNU (GNU's Not Unix) General Public License (GPL),[131] and from there the open source movement took shape with new open source software, such as GNU/Linux,[132] as well as various forms of open source licenses.[133]

Quite often, open source software is "free,"[134] and if it is sold, it is often a variant sold in conjunction with a service.[135] The obvious question to ask is what incentive is there for a programmer to invest his time in contributing to open source projects? Professors Lerner and Tirole have theorized that programmers are rewarded by a "signaling incentive,"[136] which is further divided into career concern[137] ***191** and ego gratification incentives.[138] Signaling incentives induce high motivation to produce quality work due to peer review and recognition by outsiders.[139]

The open source community is a strong opponent to the Proposed Directive in Europe,[140] and open source groups have raised a litany of concerns regarding software patents' effects on the industry and particularly on open source development. The European Commission has compiled the consultations on the Proposed Directive in a report listing the reasons why software patents should not be granted, in order of frequency mentioned: patents favor large organizations, patents are anti open source, philosophical objections,[141] software is different from other technologies, software patents provide a high risk of inadvertent infringement, copyright is adequate protection, software patents are generally unnecessary, there is a low quality of software patents, patents result in increased product costs, there is abusive use of software patents, and software patents are a threat to interoperability and standards.[142] The open source community wants a complete ban of software patents.[143] The focus of the arguments below will be on software patent effects on large versus smaller companies, not only because this is one of the largest concerns voiced in the European Commission report but also because it relates most to innovation. The effect of software patent must be viewed from a viewpoint of innovation, rather than on arguments based on cost issues[144] or quality of examination issues,[145] because a primary goal of patent law is to provide incentives to innovate.

*192 III. The Case for Computer Software Patents

The CONTU Software Subcommittee in 1978[146] and the USPTO in 1994[147] held hearings on the patentability of software. At

both hearings, opponents contested allowing software patents, and today, similar sentiments are being expressed regarding the current Proposed Directive in Europe.[148] Interestingly enough, many of the arguments have not changed, nor has the makeup of the proponents and opponents. The major proponents for software patents are large corporations, start-up companies that believe patents can help them receive seed money, and patent attorneys.[149] The major opponents in the past were individual computer programmers.[150] Surprisingly, the proportion of small and medium-sized enterprises (SMEs) in Europe that support and oppose software patents is approximately equal.[151] In addition, the open source group, which was not as visible a decade ago, has become one of the loudest voices against software patents.[152]

A. Copyrights Versus Patents

When open source advocates assert that software should not be patentable, an inevitable question to ask is whether copyright protection alone is adequate to protect computer software. As early as 1977, CONTU recognized that copyright protection was inadequate because of the dual nature of a computer program--the source code not only provides the directions for the work, it actually performs the work as well.[153] The Trade Mark, Patents and Designs Federation (TMPDF) asserts ***193** that it is possible for software to be protected by both copyrights and patents, just as many mechanical products are protected by both design rights and patents.[154]

In the software industry, the software product has two values: the source code text and the ideas. However, as stated earlier, the bulk of the value of a program is in the ideas and not the source code.[155] Once a software product has been successfully commercialized, competitors or open source groups could easily free-ride on the research invested in the application development. For example, the pharmaceutical industry is frequently cited as an example where patent protection effectively serves the goal of protecting companies from the risk of investing in innovation.[156] Due to the great costs of pharmaceutical research, brand-name companies view patents as essential to prevent free-riding and to recoup their investments.[157] Because the value of programs are in the ideas (protected by patents) and not the expression (protected by copyrights), copyright law, at least in its current state in the United States and European Union, would not be an adequate medium of protection for software.[158]

In fact, a lawsuit was recently filed in the United States that would further complicate software copyright law.[159] The complaint states that U.S. copyright law is unconstitutionally vague in defining what parts of computer programs are freely available ideas.[160] The complaint also asserts that copyright case law, which for years has struggled to define the boundaries of the idea-expression dichotomy of software, has not cured this vagueness.[161] While the lawsuit is in the early stages, the resulting decision could have a significant impact on software patents because a ***194** finding for the plaintiff would leave patents as the only source of statutory protection for software in the United States.[162]

B. Software Versus Open Source to Promote Innovation

Given that copyright law alone would not adequately protect software, the next question to answer is whether the open source movement would adequately replace patents and promote innovation in the software industry.

1. Do Software Patents Fulfill the Goals of Patent Law?

i) Purpose of Patent Protection

Patents are largely a tradeoff between society's grant of a monopoly to an inventor in return for disclosure of new ideas.[163] With regard to software, disclosure is a secondary concern because of today's rate of technological progress and ease of reverse engineering. Another primary purpose of patents, particularly in the United States, is to promote innovation by providing incentives, but not awards.[164] The idea that incentives, rather than rewards, are the driving force behind innovation has been repeated in U.S. courts regarding both copyright[165] and patents.[166] Most international legal systems provide incentives to innovate[167] by granting a "right to exclude competitors" hopefully justifying the "risk involved in investing in research ***195** and development."[168] A monopoly, normally viewed as anti-competitive, is readily given to novel innovations because, in theory, a monopoly would be limiting usage of an idea that would never have been created or disclosed otherwise.[169]

ii) Considerations Balanced by the Patent System

With the general difficulty in searching for prior art or hiring skilled patent examiners in most patent offices, computer software is more susceptible to some of the dangers of patent protection. Obvious software patents can harm competition,[170] stifle innovation,[171] increase transaction costs due to unjustified licensing,[172] increase costs in programming around the technology,[173] or cause companies to migrate to a completely different standard, thus disrupting interoperability within the industry.[174] It is important that patent offices do not grant patents to obvious patents, but no system is perfect and laws must balance whether to tilt in favor of granting a monopoly to a potentially obvious patent, or to risk less innovation in subject matter areas that are not given protection. A fitting analogy is that just as "automobile brakes . . . permit motorists to drive with greater speed," patents are brakes that inventors can apply but which encourage them to accelerate the progress of technology and the economic system as a whole.[175]

2. Large Companies Utilize Software Patents to Promote Innovation

Due to the minimal amount of capital required for software development, large research labs are not required for innovation to occur. Rather, there can arguably be a larger amount of "basement" inventors in the software field, thus not requiring patents for innovation. Although there is a subset of inventors that are willing to innovate and give their works to the public domain, the software industry as a whole, however, may need the contributions of larger research venues and those venues will surely require the investment protection that patents provide.

***196** A significant amount of software research is still conducted by traditional venues, such as large companies and universities. For example, Google, a web search company, was started by two Stanford University graduate students,[176] and the company went on to develop the now patented PageRank™ system for ranking web pages.[177] In areas such as web searching, there can be targeted research and development, as exhibited by Microsoft, Yahoo, and Google's concerted efforts to compete in and patent search technology, and, in fact, the patent wars have encouraged those companies to prolifically publish academic papers to be used as prior art.[178] Open source programmers oppose patents because they envision software patentees to be programmers that patent permutations of old algorithms and flooding the patent thicket with obvious patents. However, software research and development, like in any other industry, requires funding by companies or to universities, and without an indication of return on investment or preventing others from pirating ideas, there would be less incentive to fund new technologies.

In addition, large companies are efficiently structured to capitalize on an invention and to use patents to fund research to promote innovation. Economist Joseph A. Schumpeter emphasizes the importance of the second and third stages of technological change in order to capitalize on an invention.[179] For every $1 spent on basic research during the invention stage, $100 would be spent on development in the innovation stage, and $1000 is spent on commercialization during the diffusion stage. Therefore, while most software programmers may be able to spend the first dollar, most fail to capitalize on the rest of the process. The cost to bring a completely new product to market is extremely costly and often few companies or open source groups are motivated to undergo this process. Rather, many programmers re-code the ideas of already-developed programs.[180] The less resources a ***197** company has to spend, the less inclined they will be to divert money on support for inventions outside their core markets.[181] Large companies with more resources can take risks on basic research funding for inventions in potentially unprofitable new fields because if successful, they would not only have the resources to bring a product through the developmental stages and to a market, but patents also provide the assurance that their ideas would not be taken.

One concern in an industry where network effects create strong market positions is that large stable companies may not need to innovate. However, unless a completely monopoly exists, large software companies can still be particularly motivated to innovate. For example, Microsoft is aware that its "biggest competitor is [its] installed base . . . . '[Consumers] can sit on the existing [products]--that's a perfectly legitimate choice.'"[182] In fact, a study released at the end of 2003 showed that only 6.6% of business machines had the most current version of Windows installed.[183] Therefore, if Microsoft is to break this lull in sales, it will be required to not only innovate on old products, but invent new ones. In the 1950's, Machlup observed that "the largest research laboratories are in fact maintained by corporations with the strongest patent positions and with high and stable earnings."[184] Even today, CHI Research found that the companies with the most "highly cited" patents were also the best performers on the S&P 500 index.[185] As a result of patent protection, large companies do have the incentive to "invent" and "innovate" to create novel and useful ideas.

3. Software Patents are Beneficial to Small and Medium-Sized Enterprises (SMEs)

SMEs and individuals are understandably worried that avoiding patent infringement would be a significant cost in comparison to the cost of developing the software product itself. A major opponent of the EU Proposed Directive believes that SMEs would be burdened with legal costs incurred for every software project to determine patent infringement, licensing costs, and their own establishment of a patent portfolio.[186] However, according to the TMPDF, there is no evidence that ***198** SMEs would be at a disproportionate disadvantage, and, in fact, "patents are much more likely to be asserted against larger companies, given their greater financial exposure."[187] Large firms typically cross-license with each other at no cost because they have equally large patent portfolios.[188] The patent race is similar to a nuclear arms race where large firms will cooperate in cross-licensing in order to prevent Mutual Assured Destruction (MAD).[189] Although SMEs with smaller patent portfolios would be in a disadvantageous bargaining position, large corporations, even those with significantly large patent portfolios, cannot deter non-practicing entities (NPEs) with a MAD scenario.[190] A NPE would not be at risk of being sued because they would be outside the industry while still able to enforce their patent portfolios against a larger company. Therefore, large corporations should be willing to find fair licensing terms to ensure that SMEs are not pushed into becoming NPEs.

Moreover, a patent-granted monopoly provides SMEs with the incentive to innovate and compete with larger companies. The market position in the software industry is built on network effects, and a monopoly system would provide a perspective SME the lead time to make up for the difference in intellectual capital. Hypothetically assume that it would take 100,000 programming man-hours to create a software innovation to bring to market. Furthermore, assume an SME could only afford to allocate ten programmers to this hypothetical software product. Therefore, bringing this innovative product to market would require approximately a year. When this company releases its first product it will try and gather a network of users. In the meantime, a large company of Microsoft's stature and resources could take the innovative idea, add it onto their existing code base, release it with its own product or add it onto its Office Suite, and leverage their existing market position and user network. There would be no chance for the SME to survive, nor would there be an incentive for a venture capitalist to invest any money in the SME. Therefore, SMEs benefit greatly from patents because any "technological lead held by that SME is vulnerable to the superior resources of a larger company."[191] In fact, the study commissioned by the European Commission stated:

> ***199** It is clear that at least in the early stages of the growth of computer program related industries there was relatively little use of patents. On the other hand lack of patents will have made it easier for major players to take ideas of SMEs and independent software developers and market them without recompense to the originators. There is at least ample anecdotal evidence that this indeed occurred.[192] With patents, SMEs, such as Stac Electronics who won a $120 million judgment against Microsoft,[193] can compete against larger and more resourceful companies. Without a patent system, the SME will have to look to antitrust or competition laws to protect them from large firms, and this may ultimately require significant regulatory intrusions in the marketplace.

Furthermore, SMEs which require funding from venture capital also benefit from a patent system. For example, in a recent survey in the biotech industry, SME survey respondents listed capital access as a significant barrier to advancement[194] and patents were noted to be "essential in providing sufficient incentives for the private sector to raise capital and provide investment funding for biotech R&D."[195] Similarly, Silicon Valley dotcoms and other software companies have heavily relied on venture capital while existing as "startups."[196] Venture capitalists recognize patents as potential for novel, marketable ideas,[197] and they often insist that "companies they back have significant, patented inventions that will shield their investments from competition."[198] A strong patent regime allows investors to infuse the software industry with research and development capital, particularly by SMEs in need of outside financing.[199]

***200** 4. Dangers of an Open Source System

There is no doubt that the open source movement has created many great projects;[200] however, the open source movement is not without problems that would make it risky as the primary source of innovation or development. When there are disagreements in the design of an open source project, the result can lead to a splintering of various incompatible programs, known as "forking" of projects.[201] Furthermore, largely because of the lack of signaling incentives, open source programmers do not perform the less prestigious, but just as important, development tasks such as documentation, design of easy-to-use interfaces, technical support, and ensuring backwards compatibility.[202] Particularly because of the ego gratification incentive and lack of command hierarchy, open source projects can also exhibit fads where projects are neglected for years while others attract large numbers of programmers, leading to an "inefficient impact on the allocation of research."[203] Less attractive projects will either fade with neglect or be maintained by less skilled developers.[204] Therefore, companies that decide to rely

on an open source project bear a significant risk. If an open source project fails, a company incurs great losses, as migrating between programs disrupts day-to-day business.[205]

As described earlier, the open source commercial industry is that of services and support, and in order to provide stability and to become an "attractive alternative to corporations," an open source project requires companies willing to provide support as Red Hat does with Linux or Pervasive will for PostgreSQL.[206] However, a chicken-and-egg problem exists because for a company to want to provide support, there would need to be an indication that the open source project would be ***201** readily adopted. Therefore, commercial companies provide more stability and accountability than open source companies, particularly for certain markets, such as desktop software, which have less sophisticated users.

5. Copyright Termination on Open Source Projects

Most open source projects today are relatively young,[207] but some scholars speculate that the future possibility of widespread copyright termination may be the Achilles' heel of many open source software projects.[208] U.S. copyright law permits authors "to terminate grants of copyright assignments and licenses that were made on or after January 1, 1978 when certain conditions have been met."[209] The purpose of the section was to substitute section 24 of the 1909 Copyright Act, a provision that allowed rights to revert to the author at the time of renewal.[210] The copyright termination was a method of protecting authors from unequal bargaining power due to the "impossibility of determining a work's value until it has been exploited."[211] Thus, the provision was a safeguard "against unremunerative transfers."[212]

The copyright termination provision is an unequivocal right of reversion which can neither be waived nor contracted away;[213] thus no form of open source license would be able to protect against it. Because termination requires that notice be given,[214] projects may be able to rewrite source code of sections that are reverted to the original author; however, for extremely large projects with multiple contributors there is no telling the kind of havoc that would be created if multiple rights of termination were initiated. Moreover, if core parts of the project were reverted, it could possibly kill a project making the controversial SCO lawsuits against various companies a commonplace occurrence.[215] While companies such as IBM would be ***202** able to avoid termination rights from their employees because of the exception for works made for hire,[216] the underlying open source project would still be vulnerable, as open source projects are by definition not works for hire. Therefore, in the decades to come when open source projects become subject to the right of termination, the open source movement may experience a potentially devastating effect on the collaboration and distributed coding of the open source projects.

IV. Open Source and Patents Existing in Harmony

Open source groups, such as Eurolinux, are the loudest voice against software patents. However, if there is a complete ban of software patents, and given the current state of copyright law, open source may in fact deter innovation. The open source movement would then impose its business model on the software industry, from that of software creation to software servicing.[217] Inventions involving software affect many types of technologies which are "vital European and global industries, from automotive to telecom to medical systems."[218] An elimination of the protection and incentives software patents would provide only disadvantage the EU software industry.

Many in the open source community believe that innovation will occur either through competition or through the open source community's own innovations.[219] However, open source programs are not pervasive, and, in fact, only a few open source projects are well-known or in widespread commercial use, although the few open source programs that have a lot of support are extremely popular and already compete on a playing field where software patents exist. For example, even in the United States, Apache has a 66 percent market share in the Internet web server market[220] and Linux is steadily gaining market share.[221] New innovations from the ***203** open source community only occur in a few popular software fields where there is pronounced activity, such as operating systems, databases, and networking.[222] However, software patents not only affect interaction between the commercial software industry and the open source movement, but also interaction within the commercial software industry. In niche areas where open source groups aren't large players, the elimination of patents could mean that innovation would become stagnant or significantly decrease as companies start to appropriate each other's ideas. Therefore, both sides of the debate must analyze the breadth of the impact that open source projects truly have on the software industry.

The fact that many open source projects offer free replacements of commercial products[223] implies that many open source projects do indeed take the ideas and functionality of their commercial predecessors. Would the open source community have

had the impetus to create Linux without Unix, to create GIMP without PhotoShop, to create OpenSQL without SQL, or to create MythTV without TiVo? It is quite possible that the impetus of open source programmers is not so much a drive to innovate in a particular software niche, but rather that they want to provide a free version of an already existing software. Open source projects should be able to provide an alternative, but innovations by commercial software companies may still deserve to receive the proper protection that software patents provide.

The commercial software and open source industries provide different benefits to the consumer.[224] There is no question that the open source community has created foundational programs for the Internet or other software programs that are even used by commercial companies.[225] In addition, the open source movement not only provides an alternative to commercial products, but also produces programs that would not have been created as a commercial product.[226] On the other hand, while open source projects are geared to the more sophisticated technological elite, ***204** commercial products provide features typically geared towards the "most ignorant" users, which inevitably are a larger portion of the public.[227] Commercial software is targeted towards a larger set of society than open source. Although areas like operating systems, programming languages or interpreters, web servers, and so forth may be able to frequently attract open source programmers, commercial companies are often needed to fill the gaps in areas such as billing software which is a "staple for business but boring and unchallenging for hackers."[228] Commercial software is more user-friendly, and buying software also provides a sense of accountability. For example, a recent study showed that Microsoft fixed its security holes the quickest while also being usable for relatively unsophisticated administrators.[229] Furthermore, there is an assortment of proprietary codecs, plugins, and tools that open source would simply not be able to replace, such as Macromedia Flash or Real Networks' RealMedia files.[230] An open source advocate such as Richard Stallman would simply argue not to use those programs;[231] however, denying the adoption of standards or tools simply because they are proprietary does not seem to be an innovation-promoting stance. In addition, some commercial tools, such as Macromedia's suite of Web creation tools, may be considered better than open source alternatives,[232] and a patent system would continue to provide incentives to those private companies to continue development of their tools and standards without the worry of stolen ideas.

Commercial companies also lend credence to the acceptance of open source software. IBM, the leader in computer software patents, also heavily invests and integrates Linux into its solutions, such as selling a Linux-only mainframe.[233] IBM's support of Linux has allayed the fears of companies' concern of continued development, and this opened the potential customer base for Linux.[234] The TMPDF believes that:

> ***205** In practice a company often has a choice for any given project between investing in its own R+D, and retaining IP rights, and utilising (and enhancing) existing Open Source code, which saves R+D expense, but surrenders IP rights. Which option is taken depends upon the particular commercial circumstances, but it is vital that industry is free to continue making this choice (particularly for SMEs).[235] Commercial firms may use open source for some solutions and patented software for others, but both methods should be available to companies in order for them to make the best business decisions that will allow them to compete in the software industry.

As open source projects are embraced by the commercial software industry, there may be a trend to collectively pool patents to at least allow open source developers to freely use their ideas. Similar to Red Hat Linux, IBM recently announced that it would allow open source developers to freely use 500 software patents,[236] and Novell has vowed to use its patents to defend open-source.[237] Computer Associates followed this lead by pledging part of its patent portolio for open source use.[238] Perhaps as the open source movement grows, the open source community will amass a collective portfolio to counteract the portfolio of larger companies and be able to effectively participate in the patent arms race.

Conclusion

The arguments against computer software patents have not been tenable in the past, and many of the attacks by open source groups do not justify thwarting harmonization efforts by the European Union. To determine if software patents should exist, one must balance the incentive to innovate that software patents provide versus the frequency with which potentially obvious software patents may arise. Software patents serve to provide incentives for large companies, but they also serve to protect the innovations of SMEs. Furthermore, software patents have and can continue to exist in harmony with open source. The attacks against software ***206** patents are partially masked attacks against patent offices granting obvious patents; however, with the ongoing debate regarding software and business methods, the USPTO is making progress towards improving the quality of examination and issuance of patents in new subject matters.[239] Rather than eliminating patents as a medium of

protection, efforts should be focused on a better system and higher standards of examination to assure that only truly novel patents are being granted.[240]

Without a patent system, the software industry would have to rely on the open source movement to provide innovation.[241] The open source community may not be ideally suited to provide innovation. Nevertheless, although computer software patents can protect and help the software industry, open source still provides a good counterbalance to market dominant companies. For example, Linux's insurgence into the software operating system market is forcing Microsoft to innovate even more. While Microsoft spends $6 billion a year in research and development, it faces the risk of losing ground in its protected market.[242] Tech industry observers believe that "Microsoft may need to put more focus on creating something altogether new."[243] Companies are also able to combine the best of both worlds. For example, though Sun Microsystems announced that it would make its Solaris system open source, it also plans on protecting parts of the code with patents,[244] although ***207** it has offered 1,600 patents for use by the open source community.[245] Therefore, patents and open source should and can continue to co-exist harmoniously as they have done before.

Footnotes

a1 This article is based on information, as available to the author, as of Feb. 3, 2005. The author is a candidate for a JD and an LL.M. in International & Comparative Law, class of 2005 at Duke University School of Law. The author received his B.S. in Computer Science from Stanford University and was a former patent examiner at the USPTO in Art Unit 2176 (computer software document processing). The author would like to thank Professor David Lange for discussing his opinions on open source and also Jason Yang for his constant guidance and for looking over some early drafts of this article.

1 Lee J. Schroeder, The Harmonization of Patent Laws, in International and Comparative Patent Law 342, 342 (Graeme B. Dinwoodie et al. eds., 2002). Having a harmonized patent law creates economies of scale for companies. Kieren McCarthy, Open Source Prepares to Kiss EU Patent Ass Goodbye, The Register, June 29, 2003, at http://www.theregister.co.uk/content/4/31472.html.

2 The World Intellectual Property Organization's (WIPO) Substantive Patent Law Treaty (SPLT) is a good example. Substantive Patent Law Harmonization, World Intellectual Property Organization, at http:// www.wipo.int/patent/law/en/harmonization.htm (last visited Jan. 4, 2005).

3 Some examples include computer software, business methods, genetic engineering, plant varieties, and so forth.

4 Alan Greenspan, chairman of the Federal Reserve Board, in 2003, stated: "Over the past half century, the increase in the value of raw materials has accounted for only a fraction of the overall growth of U.S. gross domestic product. The rest of that growth reflects the embodiment of ideas in products and services that consumers value. This shift of emphasis from physical materials to ideas as the core of value creation appears to have accelerated in recent decades." McCarthy, supra note 1. The startling trend of the shift of U.S. resources from tangible to intellectual property was recognized long ago. In 1947 intellectual property was under 10% of U.S. exports. In 1986 this figure jumped to 37% and was estimated to be over 50% in 1994. Fred Warshofsky, The Patent Wars: The Battle to Own the World's Technology 6-7 (1994).

5 The European Patent Office (EPO) Technical Board of Appeals has stated before that it takes notice of developments in the United States and Japanese patent offices. Case T1173/97, IBM/Computer Programs, [2000] E.P.O.R. 219, 225 (1998). Developments in these countries' patent offices "represent a useful indication of modern trends. In the Board's opinion, they may contribute to the further highly desirable (worldwide) harmonisation of patent law." Id.

6 Proposal for a Directive 2002/047(COD) of the European Parliament and of the Council on the Patentability of Computer-Implemented Inventions [hereinafter Proposed Directive], available at http:// europa.eu.int/comm/internal_market/en/indprop/comp/com02-92en.pdf.

7 Press Release, European Commission, Patents: Commission Welcomes Council Agreement on Directive on Computer-Implemented Inventions (May 18, 2004), at http://europa.eu.int/rapid/pressReleasesAction.do?

reference=IP/04/659&format=HTML&aged=1&language=EN&guiLanguage=en.

[8] Ingrid Marson, Politicians Demand Fresh Start for Patent Directive, ZDNet UK, Jan. 10, 2005, at http:// news.zdnet.co.uk/software/0,39020381,39183538,00.htm.

[9] Committee on Legal Affairs, at http:// www.europarl.eu.int/committees/juri_home.htm (last visited Feb. 3, 2005).

[10] European Parliament JURI Committee Votes for Restart with Massive Majority, Foundation for a Free Information Infrastructure, Feb. 2, 2005, at http://kwiki.ffii.org/Restart050202En; see also JURI Votes: It's Restart, Groklaw, Feb. 2, 2005 (stating that JURI would invoke Rule 55 of the Rules of Parliament to ask for a completely new directive), at http:// www.groklaw.net/article.php?story=20050202162302114.

[11] Ingrid Marson, Patent Restart Request Ratified, ZDNet UK, Feb. 17, 2005, at http://news.zdnet.co.uk/0,39020330,39188227,00.htm. See, e.g., EU Software Patent Law Faces Axe, BBC News, Feb. 17, 2005 (stating that politicians unanimously agreed to reject the bill and that to become law the European Parliament and a qualified majority of EU states have to approve the draft wording of the bill), at http:// news.bbc.co.uk/2/hi/technology/4274811.stm.

[12] Florian Mueller, Software Patents: European Commission Negates Democracy by Declining European Parliament's Request for a Restart, NoSoftwarePatents.com, Feb. 28, 2005, at http:// www.nosoftwarepatents.com/phpBB2/viewtopic.php?t=407. The European Commission "has the sole right to propose pan-EU legislation [and stated] ... that the ministers should decide on the draft law." Reuters, EU Set for Vote on Software Patents, News.com, Mar. 4, 2005, at http:// news.com.com/EU+set+for+vote+on+software+patents/2100-1014_35599580.html? tag=nefd.top. The Directive was designated an "A-point" on the agenda, "meaning it can be approved without any debate," which would "send it back to the Parliament for a second reading, where the opportunity for amendments is much lower." Joe Fay, Euro Ministers Set to OK Patent Measure, The Register, Mar. 4, 2005, at http://www.theregister.co.uk/2005/03/04/software_paent_ directive/. Denmark was considering to have the Directive listed as a "B-item" in order to reopen discussions and possibly make amendments. Graeme Wearden & Matt Loney, Last-Gasp Attempt to Block Patent Directive, ZDNet UK, Mar. 4, 2005, at http://news.zdnet.co.uk/0,39020330,39190297,00.htm.

[13] Ingrid Marson, EU Council Endorses Patent Bill, ZDNet UK, Mar. 7, 2005, at http://news.zdnet.com/2100-3513_22-5602123.html. The EU Council adopted the Software Directive partly "for institutional reasons so as not to create a precedent which might have a consequence of creating future delays in other processes." Ingrid Marson, Software Patent Directive Adopted, ZDNet UK, Mar. 7, 2005, at http://news.zdnet.co.uk/0,39020330,39190497,00.htm.

[14] Florian Mueller, EU Council Sends a Mockery for a Common Position to the European Parliament for a Second Reading, NoSoftwarePatents.com, Mar. 7, 2005, at http://www.nosoftwarepatents.com/phpBB2/viewtopic.php?t=439. Furthermore, the Danish minister did not demand to have the Directive designated a B-item on the agenda as the Danish government had previously desired. Id.

[15] WIPO Moves Toward "World" Patent System, GRAIN, July 2002, at 3, at http://www.grain.org/briefings_files/wipo-patent-2002-en.pdf.

[16] The Results of the European Commission Consultation Exercise on the Patentability of Computer Implemented Inventions, Oct. 2000, at 31, at http:// europa.eu.int/comm/internal_market/en/indprop/comp/softanalyse.pdf. [hereinafter Results of the European Commission].

[17] Id.

[18] Id. at 36.

[19] Id.

[20] In 1999, there was a $155 billion global market, with the United States capturing 47% of this industry. Lisa Wilson, Software Development Industry Study, Small Business and Technology Development Center, Jan. 2001, at 4, at http://www.sbtdc.org/pdf/software.pdf.

[21] For example, Eastman Kodak won a lawsuit against Sun Microsystems relating to several patents infringed by Sun's Java programming language. Graeme Wearden, Kodak Wins Java Patent Suit, News.com, Oct. 4, 2004, at http:// news.com.com/Kodak+wins+Java+patent+suit/2100-1014&uscore;3-5394765.html? tag=nefd.pop. As a result, Kodak sought $1 billion in damages. Id. However, the case was eventually settled for $92 million. Stephen Shankland, Sun Settles Kodak's Java Suit for $92 Million, News.com, Oct. 7, 2004, at http:// news.com.com/Sun+settles+Kodaks+Java+suit+for+92+million/21001012_ 35401804.html?tag=nefd.top.

[22] See U.S. Const. art. I, § 8, cl. 8. The underlying purpose of intellectual property is to promote innovation through the incentive of the grant of an exclusive right. See id.

[23] Roger E. Schechter & John R. Thomas, Intellectual Property: The Law of Copyrights, Patents and Trademarks 44 (2003); 17 U.S.C. § 101 (2000) (A computer program falls in the definition of a "Literary Work" which are "works, other than audiovisual works, expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects, such as books, periodicals, manuscripts, phonorecords, film, tapes, disks, or cards, in which they are embodied.").

[24] Pub. L. 93-573, 88 Stat. 1873 (Title II) (1974) (creating CONTU whose purpose was to study copyright and make recommendations as to changes in copyright law and procedures).

[25] Final Report of the National Commission on New Technology Uses of Copyrighted Works, Chapter 3- Computers and Copyright: Copyright and Other Methods Compared, July 31, 1978 (merely stating that it is "unclear whether a patent may ever be obtained for a computer program" because of the Supreme Court decisions at the time), available at http://digital-law-online.info/CONTU/contu9.html. The recommendation resulted in the adoption of the Computer Software Copyright Act of 1980, Pub. L. 96-517, 94 Stat. 3015 (1980) (adding limitations on exclusive rights for computer programs in 17 U.S.C. § 117 and defining a "computer program" as "set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result" in 17 U.S.C. § 101).

[26] See 17 U.S.C. §102(b) (2000) (stating that "[i]n no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work").

[27] See, e.g., Whelan Assoc. v. Jaslow Dental Lab., 797 F.2d 1222, 1234-36 (3d Cir. 1986).

[28] Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1249 (3d Cir. 1983).

[29] Whelan, 797 F.2d at 1224-25.

[30] Id. at 1248.

[31] Computer Assoc. Int'l, Inc. v. Altai, 982 F.2d 693, 705 (2d Cir. 1992).

[32] Id. at 706. The court summarized its test as the following:
In ascertaining substantial similarity under this approach, a court would first break down the allegedly infringed program into its constituent structural parts. Then, by examining each of these parts for such things as incorporated ideas, expression that is necessarily incidental to those ideas, and elements that are taken from the public domain, a court would then be able to sift out all

non-protectable material. Left with a kernel, or possible kernels, of creative expression after following this process of elimination, the court's last step would be to compare this material with the structure of an allegedly infringing program. The result of this comparison will determine whether the protectable elements of the programs at issue are substantially similar so as to warrant a finding of infringement.
Id.

[33] Gregory J. Kirsch, The Changing Roles of Patent and Copyright Protection for Software, GigaLaw.com, Apr. 2000, at http://www.gigalaw.com/articles/2000-all/kirsch-2000-04-all.html.

[34] 49 F.3d 807 (1st Cir. 1995).

[35] Kirsch, supra note 33.

[36] Altai, 982 F.2d at 712. The concurring opinion in Lotus v. Borland made similar observations, stating that the "substance" of the menu structure in question had more to do with patent law. 49 F.3d at 820. (Boudin, J., concurring).

[37] Altai, 982 F.2d at 712.

[38] The historical push to only copyright the expression of software may have originated in the now rejected "printed matter" exception. See In re Beauregard, 53 F.3d 1583, 1584 (Fed. Cir. 1995) (finding that the Commissioner agreed that the printed matter doctrine was not applicable and that computer programs were patentable subject matter). From there, courts in copyright cases like Lotus v. Borland stated that aspects of the code were actually ideas that were subject to patent protection. See supra text accompanying note 34. Furthermore, in cases like Apple Computer, the courts have allowed certain aspects of software to be copyrightable. See Apple Computer Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1250-51 (3d Cir. 1983). The issue is how courts interpret software to fall in the idea-expression dichotomy, which is often traced back to Baker v. Selden, 101 U.S. 99 (1979). Schechter & Thomas, supra note 23, at 32.

[39] Apple Computer, 714 F.2d at 1250-51.

[40] Id.

[41] 450 U.S. 175 (1981).

[42] Kirsch, supra note 33. See Diehr, 450 U.S. at 176 ("A claim drawn to subject matter otherwise statutory does not become nonstatutory simply because it uses a mathematical formula, computer program, or digital computer.").

[43] See, e.g., Parker v. Flook, 437 U.S. 584, 594-95 (1978) (the applicant's use of computers for "automatic monitoring-alarming" was unpatentable because "if a claim is directed essentially to a method of calculating, using a mathematical formula, even if the solution is for a specific purpose, the claimed method is nonstatutory"); Gottschalk v. Benson, 409 U.S. 63, 71-72 (1972) (finding that "[t]he mathematical formula involved here has no substantial practical application except in connection with a digital computer, which means that if the judgment below is affirmed, the patent would wholly pre-empt the mathematical formula and in practical effect would be a patent on the algorithm itself").

[44] Diehr, 450 U.S. at 177-78.

[45] Id. at 186-87.

[46] 33 F.3d 1526 (Fed. Cir. 1994).

[47] Alappat, 33 F.3d at 1545.

[48] Id. at 1537.

[49] Id. at 1544.

[50] Id. at 1545.

[51] The court in Alappat distinguishes mathematical algorithms from patentable computer programs by "saying that certain types of mathematical subject matter, standing alone, represent nothing more than abstract ideas until reduced to some type of practical application." Gregory A. Stobbs, Software Patents 311 (1995) [hereinafter Stobbs, Software]. When taken as a whole, the computer programs must provide some type of functionality that creates a tangible result or provides a practical application. See Alappat, 33 F.3d at 1543-44 (stating that the claim must be looked at as a whole in order to determine whether it is directed to statutory subject matter).

[52] State Street Bank & Trust Co. v. Signature Fin. Group, 149 F.3d 1368 (Fed. Cir. 1998).

[53] Rather, the European Union recognizes the general sentiment against pure business method patents but recognizes that some may be allowed if they fall within the "technical character" requirement. Proposal for a Directive on the Patentability of Computer-Implemented Invention - Frequently Asked Questions, at http://europa.eu.int/comm/internal_market/en/indprop/comp/02-32.htm. Also, the EPO Board of Appeals has seemed to take a stricter stance against business methods in prior holdings. See Controlling Pensions Benefit Systems/PBS Partnership, T 0931/95-3.5.1, slip op. at 3, 10 (EPO Bd. of App. Sept. 8, 2000) (holding that a method of controlling a pensions benefits program did not "go beyond a method of doing business as such" and was excluded from patentability under Article 52(2)(c) of the EPC), available at http:// legal.european-patent-office.org/dg3/pdf/t950931eu1.pdf. Only after the pivotal State Street Bank holding did the United States allow business method patents after a history of rejecting them as patentable subject matter. Gregory A. Stobbs, Business method Patents 5 (2002) [hereinafter Stobbs, Business method]. See Hotel Security Checking Co. v. Lorraine Co., 160 F. 467, 469 (2d Cir. 1908) ("A system of transacting business disconnected from the means for carrying out the system is not, within the most liberal interpretation of the term, an art. Advice is not patentable.... 'No mere abstraction, no idea, however brilliant, can be the subject of a patent irrespective of the means designed to give it effect.'"); In re Wait, 73 F.2d 982, 983 (C.C.P.A. 1934) ("The process, when analyzed carefully, appears to comprise, in its essence, nothing more than the advertising of, or giving publicity to, offers of purchase or sale by one party, the acceptance thereof by another, and the making of a record of the transaction followed by a withdrawal of the offer. Surely these are, and always have been, essential steps in all dealings of this nature, and even conceding, without holding, that some methods of doing business might present patentable novelty, we think such novelty is lacking here."); Loew's Drive-In Theatres v. Park-In Theatres, 174 F.2d 547, 552-53 (1st Cir. 1949) (stating that "a system for the transaction of business ... however novel, useful, or commercially successful is not patentable apart from the means for making the system practically useful, or carrying it out" and the solution solved in the patent "does not seem to us to call for an exercise of the inventive faculty"). The Federal Circuit changed directions with the State Street Bank v. Signature Financial Group decision. The State Street Bank patent was for "a data processing system ... for implementing an investment structure." State Street Bank, 149 F.3d at 1370. The court held that the "transformation of data, representing discrete dollar amounts, by a machine through a series of mathematical calculations into a final share price, constitutes a practical application of a mathematical algorithm ... because it produces 'a useful, concrete and tangible result.'" Id. at 1373. The Federal Circuit also stated that many of the prior cases disallowing business method patents did so on grounds other than an explicit Business Method exception as patentable subject matter. Id. at 1375-77. Later cases focused the State Street Bank holding by declaring that the difference between mathematical algorithms and an algorithm-containing patentable invention was whether there was a "tangible, useful, result." AT&T Corp. v. Excel Communications, Inc., 172 F.3d 1352, 1361 (Fed. Cir. 1999).

[54] The European Community is in fact two legal entities: (1) the European Community (EC) and (2) the European Atomic Energy Community (Euratom). T.C. Hartley, The Foundations of European Community Law 3 (5th ed. 2003). The EC and Euratom treaties were both signed in Rome on March 25, 1957. Id. On February 7, 1992, the Treaty on European Union (TEU) was signed in Maastricht, and the European Union, a new entity, was born. Id. at 7. Since the European Union's creation there have been many

treaty developments; however, in law there are still two communities, but only one set of institutions. Id. at 9. Therefore, the developments of the European Community or European Union refer to the same governing authority whose legislation is binding on the Member States. Id. at 10.

[55] See Terence Prime, European Intellectual Property Law 248-49 (2000) (finding that computer programs require investment of resources and differences in legal protection across Member States have a direct effect on the market with respect to computer programs).

[56] Furthermore, the EPO, along with the JPO and the USPTO, have a strong influence on international doctrines, as they have formed a trilateral cooperation aimed at studying and influencing international development of patent policy. See Trilateral Web Site, at http://www.european-patent-office.org/tws/twsindex.htm (last visited Jan. 15, 2005).

[57] Convention on the Grant of European Patents (European Patent Convention), Oct. 5, 1973, art. 4(2)(a) [hereinafter EPC].

[58] Prime, supra note 55, at 175.

[59] Prime, supra note 55, at 174. Thus, the consolidating purpose and the efficiencies that result are available to any nation that chooses to participate in the Convention, which does not require membership of the European Community. Id.

[60] EPC art. 4 (stating that the European Patent Organisation, consisting of a European Patent Office (EPO) and an Administrative Council, has the task of granting European patents).

[61] Prime, supra note 55, at 176.

[62] EPC art. 64(1). See also Prime, supra note 55, at 193.

[63] Prime, supra note 55, at 177.

[64] Proposal for a Directive of the European Parliament and of the Council on the Patentability of Computer-Implemented Inventions Explanatory Memorandum 2002/0047, at 9, at http://europa.eu.int/comm/internal_ market/en/indprop/comp/com02-92en.pdf [hereinafter Explanatory Memo].

[65] Hence, many patent scholars in Europe encourage acceptance of the Community Patent Convention, which would provide a "unitary patent effective through the Community." Prime, supra note 55, at 209.

[66] Prime, supra note 55, at 174-76.

[67] Prime, supra note 55, at 175.

[68] EPC art. 52(2)(c).

[69] EPC art. 52(3).

[70] Directives, unlike regulations, were not originally intended by the EC Treaties to be given "direct effect." Hartley, supra note 54, at 206. "Direct Effect" is the principle that a "provision is applied by the national court as part of the law of the land. No rule of

national law specifically referring to it is necessary." Hartley, supra note 54, at 197. Subject to the exception that a national provision is giving effect to obligations under an international agreement, "a directly effective provision of Community law always prevails over a provision of national law." Hartley, supra note 54, at 227. Moreover, the European Court of Justice (ECJ) has increased the force of directives by holding that directives have direct effect and that "a Member State can be liable in damages for non-implementation of a directive." Paul Craig & Gráinne de Búrca, EU Law: Text, Cases, and Materials 115 (3d ed. 2003).

71 Current Situation in Europe, Foundation for a Free Information Infrastructure, Oct. 10, 2003, at http://swpat.ffii.org/news/03/situ0923/index.en.html (last visited Apr. 3, 2004).

72 Council Directive 91/250 of May 14, 1991 on the legal protection of computer programs, [1991] O.J. (L122) 42 [hereinafter Directive on the legal protection of computer programs]. Copyright is also the main form of protection for computer databases. Prime, supra note 55, at 247.

73 Directive on the legal protection of computer programs, art. 1(1). See generally Berne Convention for the Protection of Literary and Artistic Works (stating in art. 2 the scope of "literary and artistic works").

74 Directive on the legal protection of computer programs, supra note 72, Recitals para 7.

75 Directive on the legal protection of computer programs, supra note 72, art. 1(2).

76 Case T208/84, VICOM/Computer-related invention, [1987] E.P.O.R. 74 (1986).

77 Id. at 77-78.

78 Id. at 76.

79 Id. at 80.

80 As required by the Implementing Regulations to the Convention on the Grant of European Patents of 5 October 1973 as last amended by Decision of the Administrative Council of the European Patent Organisation of 13 December 2001 [hereinafter EPC Regulations], Rule 29(1) ("The claims shall define the matter for which protection is sought in terms of the technical features of the invention.").

81 VICOM, [1987] E.P.O.R. at 77.

82 Id. ("Digital filtering in general and digital image processing in particular are 'real world' activities that start in the real world (with a picture) and end in the real world (with a picture). What goes on in between is not an abstract process, but the physical manipulation of electrical signals representing the picture in accordance with the procedures defined in the claims. There is no basis in the EPC for treating digital filters differently from analogue filters.").

83 Id. at 79.

84 Id. at 80-81 ("The Board is of the opinion that a claim directed to a technical process which process is carried out under the control of a program (be this implemented in hardware or in software), cannot be regarded as relating to a computer program as such within the meaning of Article 52(3) EPC, as it is the application of the program for determining the sequence of steps in the process for which in effect protection is sought. Consequently, such a claim is allowable under Article 52(2)(c) and (3) EPC .... Generally speaking, an invention which would be patentable in accordance with conventional patentability criteria should not be

excluded from protection by the mere fact that for its implementation modern technical means in the form of a computer program are used. Decisive is what technical contribution the invention as defined in the claim when considered as a whole makes to the known art.").

[85] Case T1173/97, IBM/Computer Programs, [2000] E.P.O.R. 219, 221 (1998).

[86] Id. at 226.

[87] Id. at 227.

[88] Id.

[89] 2003 European Patent Office Guidelines for Examination, C-IV, 2.3.6 [hereinafter EPO Guidelines]. Regardless of the changes in the Guidelines for Examination, only the provisions of the EPC are binding on the Board of Appeals. EPC art. 23(3) ("In their decisions the members of the Boards shall not be bound by any instructions and shall comply only with the provisions of this Convention."). See, e.g., IBM/Computer Programs, [2000] E.P.O.R. at 225-26. A technical character or effect is generally required for patentability. See Robert Hart et al., Study Contract ETD/99/B5-3000/E/106: The Economic Impact of Patentability of Computer Programs, July 24, 2001, at 12, at http:// europa.eu.int/comm/internal_market/en/indprop/comp/study.pdf; EPC Regulations, supra note 80, at R.27(1), R.29(1). Furthermore, if "a claimed invention does not have a prima facie technical character, it should be rejected under Art. 52(2) and (3)." EPO Guidelines, C-IV, 2.3.6.

[90] Proposal for a Directive on the Patentability of Computer-Implemented Inventions - Frequently Asked Question, Feb. 20, 2002, at http:// europa.eu.int/comm/internal_market/en/indprop/comp/0232.htm.

[91] Proposal for a Directive 2002/0047(COD) of the European Parliament and of the Council on the Patentability of Computer-Implemented Inventions.

[92] Id. art. 2(1).

[93] Id. art. 4.

[94] Id. art. 5.

[95] Id. art. 6.

[96] Arlene McCarthy, Small Fry Patently Need Protection, The Guardian, June 12, 2003, available at http:// www.guardian.co.uk/online/story/0,3605,975126,00.html.

[97] The question "what is the software industry?" is related to, but not the same as, "what is computer software?" The former answers how patent protection will promote innovation, the latter answers what is protected under the patent regime.

[98] To Promote Innovation: The Proper Balance of Competition and Patent Law and Policy, FTC, (Oct. 2003), at 164-65, available at http:// www.ftc.gov/os/2003/10/innovationrpt.pdf [hereinafter To Promote Innovation].

[99] In order to get users accustomed to a certain product, much of the innovation is in the design of more intuitive interfaces for human users, what psychologists call "affordance." Pamela Samuelson et al., Symposium: Toward a Third Intellectual Property Paradigm:

Article: A Manifesto Concerning the Legal Protection of Computer Programs, 94 Colum. L. Rev. 2308, 2334 (1994). Affordance is a key principle in the computer research area of Human-Computer Interaction (HCI).

[100] Hart, supra note 89, at 30.

[101] As of 2001, Microsoft Office suite had a 90% market share. Microsoft Faces New Probe, Cnn Money, Feb. 14, 2001, at http://money.cnn.com/2001/02/14/technology/microsoft_probe/.

[102] Hart, supra note 89, at 30.

[103] In comparison to an underlying design, such as a database layout, the interface is easy to copy because a programmer only has to program a specific layout. The affordance would be manifest in the face of the software product and easy for a programmer to imitate. Samuelson, supra note 99, at 2334.

[104] See W. Brian Arthur, Increasing Returns and the New World of Business, Harv. Bus. Rev. 100, 105 Jul.-Aug. 1996 (describing that a method to capitalize on being a first-mover or having lead time is to induce interdependence on an installed base).

[105] A larger network usually implies that there will be more support and interoperability with other users. See id. at 103 (describing how high-tech products require compatibility with a network of users to induce others to use the product, particularly if it becomes a standard).

[106] Samuelson, supra note 99, at 2375.

[107] Hart, supra note 89, at 34.

[108] Hart, supra note 89, at 31.

[109] To Promote Innovation, supra note 98, at 9 (quoting Carl Shapiro, Navigating the Patent Thicket: Cross, Licenses, Patent Pools, and Standard-Setting, in 1 Innovation Policy and the Economy 119, 120 (Adam Jaffe et al. eds., 2001)).

[110] To Promote Innovation, supra note 98, at 164.

[111] To Promote Innovation, supra note 98, at 153-54; Hart, supra note 89, at 30.

[112] To Promote Innovation, supra note 98, at 154.

[113] David S. Evans et al., Software Patents and Open Source, 9 Va. J.L. & Tech. 10, 55 (2004) (arguing that many industries have cumulative research and development without suffering from innovation gridlock).

[114] The terminology of the word "innovation" is used to describe improvements, possibly design-around inventions, to a base technology. Subsequent innovations are typically incremental and require gradual research and improvement. See Murat F. Iyigun, Technology Life-Cycles and Endogenous Growth, Center for Economic Analysis, Aug. 2000, at 1, available at http://www.colorado.edu/Economics/CEA/papers00/wp00-7.pdf. The terminology of "invention" and "innovation" is different from the use of that terminology by economists like Schumpeter, who use the word "invention" to mean the generation or conception of a new idea, and "innovation" to mean the actual practice of developing the idea into a marketable product. Dan L. Burk & Mark A. Lemley, Biotechnology's Uncertainty Principle, 54 Case W. Res. L. Rev. 691, 726 n.154 (2004).

[115] Against Software Patents, The League for Programming Freedom, Feb. 28, 1991, available at http://lpf.ai.mit.edu/Patents/against-software-patents.html.

[116] Id.

[117] Hart, supra note 89, at 9. This definition is taken from a combination of the term defined by WIPO, the U.S. Copyright Code, and the EC Directive's Explanatory Memorandum. Hart, supra note 89, at 9. Some scholars have described programs as machines constructed by text. Samuelson, supra note 99, at 2320.

[118] Samuelson, supra note 99, at 2331.

[119] Samuelson, supra note 99, at 2327.

[120] Samuelson, supra note 99, at 2316.

[121] Computer Assocs. Int'l, Inc. v. Altai, 982 F.2d 693 (2d Cir. 1992). For example, well-written programs consist of code that calls functions or subroutines that have already been written and tested. This ensures improved efficiency because the same functionality for a subroutine does not have to be completely written and debugged. This also helps debugging large projects because it can allow a programmer to test smaller chunks of code. Many operating environments, such as Windows, or protocols, such as the Transmission Control Protocol (TCP), will have an Application Program Interface (API), "a set of routines, protocols, and tools for building software applications." API, Webopedia.com, at http://www.webopedia.com/TERM/A/API.html (last visited Mar. 29, 2005). Furthermore, many programming languages will have standard libraries that contain functions that are commonly used by programmers, such as functions dealing with input and output, operations on files, string manipulation, and so forth. Brian W. Kernighan & Dennis M. Ritchie, The C Programming Language 241-58 (2d ed. 1988).

[122] Perry Marshall, Innovation Displaces Invention, Perry S. Marshall & Associates, 2002, at http://www.perrymarshall.com/technical/innovation.htm.

[123] See Samuelson, supra note 99, at 2318 (describing how the most valuable part of a program is in its behavior).

[124] Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc, 797 F.2d 1222, 1231 (3d Cir. 1986).

[125] See Steven P. Schnaars, Managing Imitation Strategies 187-92 (The Free Press 1994) (describing the word processing software wars and the need to constantly come up with new features to remain dominant).

[126] There is another aspect of the open source movement, which is to generally promote companies to provide their products with their source code available to be altered. However, this aspect will not be discussed in depth because whether source code is closed or open is mutually exclusive from software programs being protected by intellectual property rights.

[127] See Simon Phipps, How Will Companies Ever Make Money Off Open-Source, JDJ, Aug. 27, 2004 (describing companies that provide compilations of open source products which customers can subscribe to, like SUN), available at http://www.sys-con.com/story/?storyid=46131&de=1.

[128] Azeem Azhar, The Microsoft Killers, Feb. 2004, at http:// www.prospect-magazine.co.uk/ArticleView.asp?accessible=yes&P_Article =12404. See also To Promote Innovation, supra note 98, at 47.

[129] Stallman is the founder of the Free Software Foundation. http:// www.gnu.org/fsf/fsf.html (last visited Apr. 26, 2005).

[130] Azhar, supra note 128.

[131] Also known as copyleft, the GPL license is available at http:// www.gnu.org/copyleft/gpl.html.

[132] Linus Torvalds wrote the source code for Linux, the kernel to the GNU operating system, and GNU/Linux is distributed under the GPL license. Azhar, supra note 128; Richard Stallman, Linux and the GNU Project, GNU.org, at http://www.gnu.org/gnu/linux-and-gnu.html (last visited Jan. 10, 2005).

[133] For a list of approved open source licenses approved by the Open Source Initiative (OSI), visit http://www.opensource.org/licenses/. There are also an assorted number of articles describing the open source or free software paradigm. See Stephen M. McJohn, The Paradoxes of Free Software, 9 Geo. Mason L. Rev. 25 (2000); Dennis M. Kennedy, A Primer on Open Source Licensing Legal Issues: Copyright, Copyleft, and Copyfuture, 20 St. Louis U. Pub. L. Rev. 345 (2001); Mathias Strasser, A New Paradigm in Intellectual Property Law? The Case Against Open Sources, 2001 Stan. Tech. L. Rev. 4 (2001).

[134] Programs are given for "free," meaning without money; however, under the FSF definition, the word "free" references freedoms, such as freedom to run the program, freedom to study the program, freedom to redistribute copies, and freedom to improve. J.T. Westermeier, Open Source Software, 801 PLI/Pat 421, 428 (2004).

[135] For example, Red Hat Linux provides software updates, systems management, and support for the open source Linux project. See Red Hat, at http://www.redhat.com (last visited Mar. 25, 2005). See also Tim O'Reilly, Lessons from the Lay offs at Linuxcare, Linux Dev Center, May 9, 2000 (describing the open source business model as a service business), at http:// www.linuxdevcenter.com/pub/a/linux/2000/05/09/lessons.html.

[136] Josh Lerner & Jean Tirole, The Simple Economics of Open Source, NBER Working Paper No. w7600 (2000), at 15, available at http:// www.hbs.edu/research/facpubs/workingpapers/papers2/9900/00-059.pdf.

[137] Id. at 14 ("refers to future job offers, shares in commercial open source-based companies, or future access to the venture capital market."). Essentially, programmers can exhibit their work by having their code integrated into the larger project. Then their work is seen by other coders, and it is like having a work portfolio distributed to everyone who views the code.

[138] Id. ("stems from a desire for peer recognition.").

[139] See id. at 16-17.

[140] The group No Software Patents, which believes Linux and open source can free software development in Europe, and open source companies are backing a campaign against the Proposed Directive. No Software Patents!, NoSoftwarePatents.com, at http:// www.nosoftwarepatents.com/en/m/intro/index.html (last visited Jan. 15, 2005); Laura Rohde, Open-Source Backers Start European Antipatent Campaign, IDG News Service, Oct. 22, 2004, available at http://www.pcworld.id g.com.au/index.php/id;1632342595;fp;2;fpid;1. The open source community has been very active recently, particularly in Europe, in fighting software patents. The Eurolinux Alliance and the Foundation for a Free Information Infrastructure (FFII) are European coalitions dedicated to promoting open source and protesting software patents. On April 4, 2004, the FFII conducted a protest in Brussels against software patents. http://demo.ffii.org/demo0502/.

[141] Referring to the general belief that software patents are morally wrong.

[142] Results of the European Commission, supra note 16, at 6.

[143] Linus Torvalds et al., Appeal to the EU Council, NoSoftwarePatents.com, Nov. 23, 2004, available at http:// www.nosoftwarepatents.com/en/m/intro/app0411.html.

[144] Many of the complaints against software patents, when broken down to the core argument, are that the open source community, which consist of individual developers, cannot afford many of the costs inherent in a patent system. There are costs in prosecuting and obtaining patents (either offensive or defensive), maintenance fees on patents, obtaining opinion letters, licensing patents, conducting patent searches, etc. which are difficult without a large source of finances. However, these arguments are not related to the arguments of whether patents promote innovation, rather they are concerned with costs which would be inherent in having any administrative system. As will be explained later, large software companies happen to be better suited for pooling financial resources, but this is true of Wal-Mart being able to pool its purchasing power over small family-sized stores.

[145] Another argument that is not addressed is whether there are too many obvious patents granted. However, the core of this argument is that the examination process is poorly conducted, rather than proving that software patents inherently do not promote innovation.

[146] Stobbs, Software, supra note 51, at 30.

[147] Stobbs, Software, supra note 51, at 30-31, Appendix A.

[148] See FFII: Software Patents in Europe, at http:// swpat.ffii.org/intro/index.en.html (last visited Feb. 28, 2005).

[149] Stobbs, Software, supra note 51, at 31.

[150] Stobbs, Software, supra note 51, at 31.

[151] The Results of the European Commission Consultation Exercise on the Patentability of Computer Implemented Inventions, Oct. 2000, at http:// europa.eu.int/comm/internal_market/en/indprop/comp/softanalyse.pdf, at 14.

[152] Id. at 12-13.

[153] Pal Asija, How to Protect Computer Programs 38 (1983) (citing Software Sub-Committee Report and Additional View, 1977, CONTU Washington, D.C. 20558).

[154] Letter from the Trade Marks Patents and Designs Federation (TMPDF) to the European Consultation Commission Dec. 14, 2003, available at http:// europa.eu.int/comm/internal_market/en/indprop/comp/tmpdf.pdf [hereinafter TMPDF letter] (The TMPDF represents UK industrial companies and works closely with the Confederation of British Industry (CBI)).

[155] See supra text accompanying note 133.

[156] To Promote Innovation, supra note 98, at 110.

[157] To Promote Innovation, supra note 98, at 118.

[158] See John T. Soma, et al., Software Patents: A U.S. and E.U. Comparison, 8 U. Balt. Intell. Prop. L.J. 1, 43 (2000) (stating that the

EPO's view to software patentability is related more to "international pressures to remain competitive with nations like the United States that, since 1981, have had a more preferential attitude towards software patents, and a realization that patent type protection is needed to protect software developers because copyright protection is proving inadequate").

159 Lawsuit Filed to Prohibit Copyright Protection of Software, Cnn.Com, Dec. 14, 2004, at http://www.unh.edu/oipm/copyrightsoftware.PDF.

160 Aharonian v. Ashcroft, Complaint for Declaratory Judgment at 5, (N.D. Cal.), filed Dec. 7, 2004, available at http://www.iplaw-quality.com/lawsuit/complaint.pdf.

161 See id. at 11-16 (discussing many of the copyright cases found in the U.S. Copyright law section of this paper).

162 The plaintiff, intellectual property consultant Greg Aharonian, argues that software is adequately protected by patents. Lawsuit Filed to Prohibit Copyright Protection of Software, supra note 159. Progress for the case from the viewpoint of the plaintiff can be found at his website. See Greg Aharonian, Is Software Copyright Unconstitutional?: A Lawsuit to Resolve Thirty Years of Vagueness, available at http://www.iplaw-quality.com/lawsuit/ (last visited Jan. 13, 2005).

163 See, e.g., Bonito Boats, Inc. v. Thunder Craft Boats, Inc., 489 U.S. 141, 157 (1989) (the ultimate goal of federal patent policy is "public disclosure and use"). An important concern of patent law is determining whether an invention can be kept as a trade secret longer than the period for which patents are granted. See Fritz Machlup, An Economic Review of the Patent System, in The International Intellectual Property System: Commentary and Materials 241 (Frederick Abbott et al. eds., 1999). Disclosure of the patent is not only advantageous to share those ideas that are "secret," but it also may "give 'ideas' to technicians in other industries who would not, as a rule, go out of their way to 'find' the technical information in question, but may be glad to take a hint when it is 'thrown at them through publication in the official gazette." Id.

164 See U.S. Const. art. I, § 8, cl. 8. If innovation in a subject matter were to occur regardless of a reward, then a patent regime should not protect that subject matter.

165 E.g., Feist Publications, Inc. v. Rural Telephone Service Co., 499 U.S. 340, 349 (1991) ("The primary objective of copyright is not to reward the labor of authors, but '[t]o promote the Progress of Science and useful Arts."); Schechter & Thomas, supra note 23, at 18 ("The copyright law has often been rationalized in terms of the 'incentive theory."').

166 E.g., Sinclair & Carroll Co. v. Interchemical Corp., 325 U.S. 327, 330-31 (1945) ("The primary purpose of our patent system is not reward of the individual but the advancement of the arts and sciences. Its inducement is directed to disclosure of advances in knowledge which will be beneficial to society; it is not a certificate of merit, but an incentive to disclosure.").

167 Schroeder, supra note 1, at 49.

168 Philips Electronics N.V. v. Ingman Limited, [1998] 2 C.M.L.R. 839, 854 (Ch. 1998).

169 Hart, supra note 89, at 28.

170 To Promote Innovation: The Proper Balance of Competition and Patent Law and Policy: FTC Executive Summary, FTC, Oct. 2003, at 3, available at http://www.ftc.gov/os/2003/10/innovationrptsummary.pdf.

171 Id. at 5.

172 Id. at 6.

[173] See id.

[174] See Aaron C. Chatterjee, Europe Struggles over Software Patents, IEEE Spectrum Online, Sept. 9, 2004 (citing the fears of several software groups that software patents may hinder interoperability of computing systems), available at http://www.foley.com/publications/pub_detail.aspx?pubid=2261.

[175] Machlup, supra note 163, at 243.

[176] Google History, Google.com, at http:// www.google.com/corporate/history.html (last visited Jan. 13, 2005); Sergey Brin's Home Page, at http://www-db.stanford.edu/~sergey/ (last visited Jan. 13, 2005).

[177] Our Search: Google Technology, Google.com, at http:// www.google.com/technology/index.html (last visited Jan. 13, 2005); Stefanie Olsen, Google Lands Web Search Patent, News.com, Feb. 26, 2003, available at http://news.com.com/2100-1024-986204.html?tag=nl. Since then, Google has also patented other search innovations. Stefanie Olsen, Overture to a Patent War?, News.com, July 18, 2003, at http://news.com.com/2100-1024_3-1027084.html. However, now Microsoft is hoping to enter the web search wars, and is already starting to imitate many of the features offered by Google. Sebastian Rupley, Microsoft Tool Seeks to Compete, ABC.com, Jan. 28, 2003, at http:// abcnews.go.com/sections/scitech/ZDM/msn_google&uscore;pcmag_040128.html; Stefanie Olsen, Microsoft's Aggressive Search Plans Revealed, News.com, Mar. 26, 2004, at http://news.com.com/2100-1025-5180519.html?tag=nl.

[178] Stefanie Olsen, Patents Raise Stakes in Search Wars, News.com, Feb. 25, 2004, at http://news.com.com/2100-1024-5165272.html.

[179] Evan I. Schwartz, Sparking the Fire of Invention, MIT's Mag. Innovation Tech. Rev., May 2004, at 36.

[180] For example, just searching through sourceforge.net will show many open source projects, many of which are open source versions of proprietary applications.

[181] Schwartz, supra note 179, at 34.

[182] Jay Greene, Microsoft's Midlife Crisis, BusinessWeek, Apr. 19, 2004, at 92.

[183] Ina Fried, Users Cling to Old Microsoft Operating Systems Dec. 12, 2003, available at http://news.com.com/2100-1016_3-5121458.html? tag=nefd&uscore;top.

[184] Machlup, supra note 163, at 244. IBM, the number one patent recipient in the private sector for the 11th year in a row, also has a very healthy research focus. See IBM Research, IBM (listing various research initiatives) at http://www.research.ibm.com/ (last visited Jan. 13, 2005).

[185] Schwartz, supra note 179, at 37-38.

[186] The CEA-PME (Confédération Européenne des Associations de Petites et Moyennes Entreprises), representing 500,000 SMEs of 19 European countries. Walter Grupp, CEA-PME Statement on the Software-Patent-Directive, Sept. 16, 2003, available at http://swpat.ffii.org/papers/eubsa-swpat0202/ceapme0309/index.en.html.

[187] TMPDF letter, supra note 154.

[188] See Alexis Grandemange, Patents, PageBox, at 45 (stating that strong patent owners against other strong patent owners result in cross-licensing and open the market to their competition, thus only losing in opportunity costs, while strong owners against weaker owners can extract licensing fees) at http://www.pagebox.net/patent.pdf (last modified Mar 17, 2005).

[189] (MAD is a term coined to describe nuclear deterrence, in which "a full scale use of nuclear weapons by one of two opposing sides would result in the destruction of both the attacker and the defender.") Mutual Assured Destruction, Wikipedia, at http://en.wikipedia.org/wiki/Mutual_assured_ destruction (last visited Jan. 13, 2005).

[190] To Promote Innovation, supra note 98, at 101; Mark A. Lemley, Essay: Rational Ignorance at the Patent Office, 95 Nw. U.L. Rev. 1495, 1505 (2001).

[191] TMPDF letter, supra note 154.

[192] Robert Hart, et al., Study "The Economic Impact of Patentability of Computer Programs," European Commission, July 24, 2001, at http:// europa.eu.int/comm/internal_market/en/indprop/comp/studyintro.htm.

[193] McCarthy, supra note 1.

[194] A Survey of the Use of Biotechnology in U.S. Industry, U.S. Dept. of Commerce, Oct. 2003, at 92, at http:// www.technology.gov/reports/Biotechnology/CD120a_0310.pdf.

[195] Id. at 93.

[196] For example, even after the dot-com bust, start-ups in Silicon Valley still received $1.6 billion of venture capital funding in 2003. After the Drought-Venture Capital, Economist, Apr. 3, 2004, at 76.

[197] Particularly in software, there is less need for patents to serve as an incentive for investment, than on the need for patents as sources of finance for such investment. See generally Machlup, supra note 163, at 244 (noting that patents are needed to receive financing for further investment). Furthermore, "[v]enture capitalists use client patents ... as evidence that the company is well managed, is at a certain stage in development, and has defined and carved out a market niche." Lemley, supra note 190, at 1505-06.

[198] Schwartz, supra note 179, at 36.

[199] Hart, supra note 89, at 32.

[200] Linux, Apache and many others are used by industry, but a relatively comprehensive list of open source projects exists at http:// sourceforge.net/.

[201] Hart, supra note 89, at 7. Even recently, forking was cited as a reason why Sun did not want to make the Java programming language open source. Martin LaMonica, Sun Reluctant to Make Java Open Source, News.com, Mar. 17, 2004, at http://news.com.com/2100-7344-5173427.html. (Sun prefers instead to continue with its method of changing Java through the Java Community Process (JCP)). Martin LaMonica, IBM Urges Sun to Make Java Open Source, News.com, Feb. 26, 2004, at http://news.com.com/2100-1007-5165427.html. Furthermore, variants can have a negative impact because the different versions are usually not cross-compatible. On the other hand, fragmentation is not always bad because it allows different versions to cater to different segments. Carolyn A. Kenwood, A Business Case Study of Open Source Software, MITRE, (July 2001), at xiv, at http://www.mitre.org/work/tech_papers/tech_papers&uscore;01/kenwood_ software/kenwood_software.pdf.

[202] Lerner & Tirole, supra note 136, at 19.

[203] Lerner & Tirole, supra note 136, at 15.

[204] Kenwood, supra note 201, at xiv.

[205] See Joanna Glasner, MySQL Profits From Open Source, WIRED, Feb. 23, 2004, at http://www.wired.com/news/infostructure/0,1377,62369,00.html?tw=wn_ tophead_6.

[206] Martin LaMonica, Open-Source Database Gets Backing, News.com, Jan. 10, 2005, at http://news.com.com/Open-source+database+gets+backing/2100-7344_3-5519933.html?tag=nefd. top.

[207] Stallman's FSF was founded in 1985, hence many open source projects could not be much older than twenty years old.

[208] Discussions with David L. Lange, Professor, Duke University School of Law, Sept. 2004.

[209] Termination of Transfers and Licenses Under 17 U.S.C. §203, United States Copyright Office, available at http:// www.copyright.gov/docs/203.html (last visited Jan. 4, 2005). See 17 U.S.C. § 203 (2000).

[210] H.R. Rep. 94-1476, at 123 (1976), reprinted in 1976 U.S.C.C.A.N. 5659, 5739. See 17 U.S.C. § 24 (1909).

[211] H.R. Rep. No. 94-1476, at 123.

[212] Id.

[213] Id.

[214] See 17 U.S.C. § 203(a)(4).

[215] SCO has filed several lawsuits against companies running Linux or with business related to Linux projects, such as IBM, Novell, AutoZone, and DaimlerChrysler, for misappropriation of its intellectual property, in particular, its UNIX copyrights. Intellectual Property, SCO, at http:// www.sco.com/company/intellectual_property/ (last visited Jan. 4, 2005); SCO Controversy Timeline, Linux Online, at http:// www.linux.org/news/sco/timeline.html (last visited Jan. 4, 2005).

[216] 17 U.S.C. § 203.

[217] McCarthy, supra note 96.

[218] Andy Reinhardt, Commentary: Inventing A Better Patent Law, BusinessWeek, Dec. 22, 2003, available at http://www.businessweek.com:/ print/magazine/content/03_51/b3863109_mz022.htm?mz.

[219] In a survey conducted by the UK Patent Office, respondents asserted that there is no shortage of innovation and in fact that open source has stimulated innovation. Annex C: Summary Analysis of Respondents' Views, UK Patent Office, at http://www.patent.gov.uk/about/consultations/annexc.htm (last updated Jan. 11, 2005).

[220] McCarthy, supra note 96.

[221] On servers, Linux has a market share of just below 20% in 2004 and expected to rise to over 28% by 2008. Margaret Follmer et al., The Linux Marketplace - Moving From Niche to Mainstream, Dec. 14, 2004, at 8, at http:// www.osdl.org/docs/linux_market_overview.pdf. On desktops its market share is about 3.8% in 2004 with an expected rise to 7% by 2008. Bruce Gain, Does Linux Have Game?, Tom's Hardware Guide, Dec. 30, 2004, at http:// www.tomshardware.com/business/20041230/index.html.

[222] This can be inferred by looking at some of the open source software leaders that are commonly cited in the news, such as Apache HTTP Server, Linux, Mozilla, MySQL, etc.

[223] See generally, Jem Matzan, Your Software Rights or the Best Tools: Often a Sad Choice, NewsForge, Jan. 7, 2005 (describing how open source provides alternatives to proprietary software, but these open source alternatives are not necessarily better programs), at http:// trends.newsforge.com/article.pl?sid=05/01/05/1842204&tid=138.

[224] See generally Craig A. James, The Care and Feeding of Foss (or, The Lifecycle of Software Technology) (describing the appropriate place for Free Open Source Software to coexist in a commercial software industry and why both commercial software and open source software have their own benefits), at http://www.moonviewscientific.com/essays/software_lifecycle.htm#commerce (last visited Jan. 2, 2004).

[225] See, e.g., Lerner & Tirole, supra note 136, at 18 n.12 (noting that commercial companies use open source tools to fix bugs).

[226] For example, many of the peer-to-peer file-sharing programs are generally not created as commercial products, nor would they be marketable given the nature of the program, though they have high levels of contributors. For activity levels of open source projects, view the sourceforge.net webpage. http://sourceforge.net (last visited Apr. 26, 2005).

[227] Lerner & Tirole, supra note 136, at 8.

[228] Azhar, supra note 128.

[229] Steven J. Vaughan-Nichols, Linux vs. Windows: Which Is More Secure?, eWeek.com, Mar. 30, 2004, at http:// www.eweek.com/article2/0,1759,1557749,00.asp.

[230] See Interview by Timothy R. Butler with Richard M. Stallman, Founder, Free Software Foundation (Aug. 13, 2003) (asking Stallman what should be done about the proprietary files to which Stallman responds that users should pressure browsers not to use those proprietary files), at http:// www.ofb.biz/modules.php?name=News&file=article&sid=260.

[231] Id.

[232] Matzan, supra note 223.

[233] See Linux Servers, IBM, at http://www-1.ibm.com/servers/eserver/linux/home.html?c=serv-ersintro&n=Linux2001&t=ad (last visited Jan. 13, 2005); Linux at IBM, IBM, at http://www-1.ibm.com/linux/ (last visited Jan. 13, 2005).

[234] Azhar, supra note 128.

[235] TMPDF letter, supra note 154.

[236] Stephen Shankland, IBM Offers 500 Patents for Open-Source Use, News.com, Jan. 10, 2005, at http:// news.com.com/IBM+offers+500+patents+for+open-source+use/2100-7344_3-5524680.html. However, many question the motivation of IBM's actions, considering that they own a considerable amount of software patents and still consider IBM's arsenal of software patents, and in fact all software patents, to be a threat to open source development. See Robert McMillan, Torvalds Joins in Anti-Patent Attack, TechWorld, Feb. 2, 2005 (discussing the general problems of software patents despite the recent donation by IBM and Sun), at http:// www.techworld.com/applications/news/index.cfm?NewsID=3059; see also infra text accompanying note 245 (regarding the discussion of donation of software patents to the open source community).

[237] Martin LaMonica & Stephen Shankland, Novell Vows Patent Defense of Open Source, News.com, Oct. 12, 2004, at http:// news.com.com/Novell+vows+patent+defense+of+open+source/21007344_3-5406571.html.

[238] Martin LaMonica, CA Readies Patent Pledge, News.com, Feb. 18, 2005, at http://news.com.com/CA+readies+patent+pledge/2100-7344_3-5582697.html? tag=nefd.top.

[239] Automated Financial or Management Data Processing Methods (Business Methods), United States Patent & Trademark Office, at http:// www.uspto.gov/web/menu/busmethp/index.html (last visited Jan. 4, 2005). See also Patent Business Methods, United States patent & Trademark Office (listing various examination guidelines and tools specifically directed to the business method art units), at http://www.uspto.gov/web/menu/pbmethod/ (last visited Jan. 4, 2005).

[240] See Evans, supra note 113, at 41-43 (stating that USPTO reforms may be able to rectify problems in the patent process).

[241] The alternative would be to create a sui generis protection system primarily for software; however, though this proposal has been suggested before, experts in the U.S. software industry have generally argued against a sui generis system of protection. See Victor Sibel, Transcript, at http:// www.jamesshuggins.com/h/tek1/software_patent_ibm.htm (last visited Jan. 4, 2005) (stating that creating a new system would disadvantage U.S. companies against competitors in the European Union and Japan because of the loss of patent priorities). Generally, a sui generis system would also create higher transaction costs because a whole system of new procedural requirements would have to be created and implemented, rather than merely building on the history and experience of established patent offices. There would also be no guarantee that the system would work, and it would take years for problems in the system to be fixed, years which could be spent improving the patent examination process for software patents. In addition, creating a sui generis system would set a bad precedent as industries of new subjects matters may also ask for specialized systems of protection. Ultimately, although fitting subject matters within the bounds of copyright or patents may not be ideal, it is far better than creating a new system of protection for each new industry.

[242] Microsoft's Midlife Crisis, supra note 182, at 96.

[243] Microsoft's Midlife Crisis, supra note 182, at 92.

[244] Stephen Shankland, Sun Plans Patent Protection for Open-Source Solaris, News.com, Nov. 18, 2004, at http:// news.com.com/Sun+plans+patent+protection+for+open-source+Solaris/2100-7344_3-5456451.html?tag=nefd.lede.

[245] Press Release, Sun Microsystems, Sun Grants Global Open Source Community Access to More than 1,600 Patents (Jan. 25, 2005), at http:// www.sun.com/smi/Press/sunflash/2005-01/sunflash.20050125.2.html. The recent donations of software patents to the open source community has caused much debate and speculation regarding the role of software patents and its impact on the open source community, as well as, discussion regarding the interaction of open source licenses. Though Sun offered 1,600 patents which encompass its OpenSolaris for use under it s Community Development and Distribution License (CDDL), some assert that Sun did not truly donate its patents to the open source community at large, that is, any OSI approved license, as IBM did. David Berlin, Will Sun's 1600 Patents Suck the Life Out of Linux?, ZDNet Blogs, Jan. 27, 2005, at http://blogs.zdnet.com/BTL/index.php?p=978.

13 TXIPLJ 171