

# RECONSIDERING COPYRIGHT PROTECTION FOR SOFTWARE

Dina Koutouki\*

## Introduction

As we approach the year 2000, technology is the topic at the forefront of many discussions concerning the next millennium. Year 2000 computer bug aside, many in the media and in general society are becoming more and more aware of the role technology plays in their daily lives. The general public is showing an increased interest in the way that technology is developed, used and distributed.<sup>2</sup> In the legal sphere, there is a second aspect to this discussion: how to ensure the legal protection of technological innovations. Of course, the answer differs depending on the innovation in question. In the case of software programs the agreed upon answer appears to be copyright.<sup>3</sup> Copyright protection for software exemplifies the relationship between technology, the law and the public. The exponential increase in the use of software has had a corresponding increase in the amount of litigation, scholarly writings and mainstream media articles devoted to this subject.

One reason for the attention paid to the world software is that software are the means by which we use numerous devices and access useful information. Considering that we live in an era where information is a valuable commodity, software have been interwoven into our society to the point that very little can be done without them. Thousands of everyday tasks and devices, everything from diagnostic medicine and airplanes to teaching elementary students and balancing chequebooks, depend on computer software. Access to information and the means by which to retrieve information, as well as, the proper functioning of software dependent devices are of interest to a wide segment of the population. With the advent of the Internet our

---

\*Dina Koutouki is an LLD candidate at the University of Ottawa and a Lecturer at the Carleton University Department of Law. She can be reached at kalesmeno@hotmail.com.

<sup>2</sup> A brief look at recent newspapers, television programs, films and other forums of popular culture makes this assertion evident.

<sup>3</sup> Foley, Hoag & Eliot, LLP, "Recent Court Decision Increases Opportunities for Patenting Software in the United States", Patent Update, July 1998. The article concerns the US Federal Circuit Court of Appeals case of *State Street & Trust Co. v. Signature Financial Group, Inc.* involving the validity of the patenting of a data processing system for implementing a mutual fund investment structure. The Federal Circuit upheld the validity of such a patent by rejecting the business methods exception to patentability and went on to say that Congress intended "anything under the sun made by man" to be patentable, including software.

This paper does not deal with the patent issue, except insofar as it is evidence of overprotection of software and the functional nature of software.

dependence on software continues to grow. Banking, communications, commerce, information retrieval, entertainment etc. are all conducted with the essential help of Internet software. The overwhelming influence that software have on the way we work, play, are educated, and use information has to be considered when deciding on a method of legal protection for these products. Our society has a very strong interest in having access to the latest, best quality and most efficient versions of these tools. They assist us in becoming not only a productive but also a meaningful society.

This paper will argue that copyright protection in conjunction with the extensive technological protection methods available to software will lead to overprotection, and consequently to the undermining of the rights of users. Furthermore, it will be argued that user rights are not only affected by lack of access to software, but also by the lack of new, innovative products. This occurs because competitors are not able to build on previous technology due to legal protection through intellectual property laws. Since software are works which require access to previous know-how to produce new products, overprotection causes the stifling of innovation, depriving users of products that would otherwise be available.

Another relevant point is that software have certain unique properties that make them poor candidate for copyright protection. It will be shown that software are functional tools and copyright provides excessive protection for such works. Furthermore, this paper will discuss the dilution of authors' rights due to the inclusion of software developers as authors. It will be argued that the needs and aspirations of these developers are substantially different from those of traditional authors. Any dilution of the rights that traditional authors possess will be a detriment to the arts and to society in general.

Finally, although it appears that copyright will not be replaced any time soon as the legal method of protection for software, it is nonetheless a controversial topic worthy of further discussion. The paper will conclude that the importance of software as functional tools cannot be overstated. Given that we live in an information age, tools that store, retrieve or otherwise manage information should be subject to legal rules that take into account the needs of the wider society. The best way to ensure quality products and the protection of user rights is to discourage monopolies and expand the fair dealing provisions of the *Copyright Act*.<sup>4</sup>

### **Copyright and Technological Innovations**

There has always been tension, in the field of copyright, between restricting access to a work for the benefit of an author or publisher and access to the information by a

---

<sup>4</sup> R.S.C. 1985, c. C-42 s. 2.

user or the public. Copyright inherently constructs a monopoly for the purpose of ensuring that an author or publisher is financially rewarded for her effort. At the same time copyright was formulated to also ensure the promotion of ideas and information for the greater good of a society.

Copyright, as we know it today, is essentially a product of an earlier information technology revolution, one created by the invention of the Gutenberg Press. This printing press first appeared in England in 1476.<sup>5</sup> Soon after, as early as 1529, laws were put in place to deal with dissident material circulating in England. The press had made it much easier to reproduce and hence disseminate such material.<sup>6</sup> The Charter of the Stationer's Company, enacted in 1556, gave 97 printers exclusive right to own a printing press and the tools of printing. As well, this group was given the exclusive right to enforce its monopoly by allowing them to burn books printed by its competition and to imprison anyone owning a press or found to be engaged in printing.<sup>7</sup>

This monopoly, which ensured that entry into the publishing industry and the works to be published were strictly controlled and that the right to publish a work was perpetual, was abolished in 1694.<sup>8</sup> Publishers, however, being well aware of the benefits of having a monopoly, began lobbying for a copyright regime that would fulfill the role previously held by the Stationer's company.

### *The Statute of Anne*

The *Statute of Anne*, otherwise known as the *Copyright Act* of 1709, is said to be the world's first copyright Act.<sup>9</sup> The *Statute* reflected the desire of legislators to avoid the situation which existed under the Charter of the Stationer's Company. Due to new limitations imposed by the *Statute of Anne*, the printers realised that their monopoly was in serious jeopardy. The public seemed much more sympathetic to the plight of the author than the demise of the printers' monopoly. Because of this, the publishers

---

<sup>5</sup>Harold G. Fox, *The Canadian Law of Copyright and Industrial Design*, (Toronto: 1967).

<sup>6</sup>Rebecca Moore Howard, "Some Events and Ideas in the History of Authorship in the West". [Http://departments.colgate.edu/diw/RMH/ChronAuth.html](http://departments.colgate.edu/diw/RMH/ChronAuth.html) Also: John Feather, "From Rights to Copies to Copyright: The Recognition of Author's Rights in English Law and Practice in the Sixteenth and Seventeenth Centuries" (1992) *Cardozo Arts & Ent. L. J.* 455.

<sup>7</sup>Howard B. Abrahms, "The Historic Foundations of American Copyright Law" (1983) 29 *Wayne L. Rev.* 1119.

<sup>8</sup>*Ibid.*

<sup>9</sup>Brian A. Carlson, "Balancing the Digital Scales of Copyright Law" (1997) 50 *SMU L. Rev.* 825.

aligned themselves with the authors and began lobbying for authors' rights, this of course, indirectly benefited them as well.<sup>10</sup>

The *Statute of Anne* had some effect in discouraging monopolies, and thus met its primary objective. One of the shortcomings of the *Statute* was that it "constituted the author as well as the publisher with legal standing".<sup>11</sup> A British literary historian said the following in reply to criticism that he excluded copyright in his narrative:

The various copyright acts and international treaties... made little difference to the lives of ordinary writers. The extension of the copyright period under the Acts of 1814 and 1842 for example, had no effect on the majority of writers because they rarely owned the copyright to their books. Publishers, however, did benefit; they were given more time in which to exhaust the copyrights they had bought from their authors. James Grant sold the copyrights of his popular historical novels to Routledge for between 100 pounds and 250 pound a time. Between 1856 and 1882 Routledge sold 100,000 copies of Grant's *Romance of War*: no wonder Grant described authorship as a "hopeless treadmill".<sup>12</sup>

This unfairness is also illustrated by Edward Lear who sold the copyright in his *Book of Nonsense* for 125 pounds and saw it go to 19 editions in his lifetime without receiving "a single penny more in royalties."<sup>13</sup>

It seems accurate to say that copyright law, in England, was formulated to balance out the rights of various publishers. Authors were used in order to make it more palatable to the public. David Vaver in his discussion of intellectual property states the following:

The first myth is that copyright is designed to protect authors. In locating itself around the central character of the author copyright law is astute.... Most copyrights and patents belong not to individual creators and inventors but to the firms that employ them. One can go further. Copyright law did not grow up to protect authors. There were indeed some big names associated with the first copyright statute, the Statute of Anne of 1710: Swift, Defoe, Addison. But consider how Lord Camden described the scene surrounding the passage of the Statute:

In the year 1708 they came up to parliament in the form of petitioners, with tears in their eyes, hopeless and forlorn; they brought their wives and children to

---

<sup>10</sup>Edward Earle, "The Effect of Romanticism on the 19<sup>th</sup> Century Development of Copyright Law" (1991) 6 *Intell. Prop. J.* 269.

<sup>11</sup>*Ibid.*

<sup>12</sup>N. Cross, *The Common Writer: Life in Nineteenth Century Grub Street* (Cambridge: Cambridge University Press, 1985).

<sup>13</sup>David Vaver, "Some Agnostic Observations on Intellectual Property" (1990) 6 *Intell. Prop. J.* 125.

excite compassion, and induce parliament to grant them a statutory security. They obtained the Act. And again and again sought for a further legislative security.

Who were "they"? Authors? No. It was the stationers, the publishers and retailers of books of the day, of whom Camden speaks. Eighteenth century authors were not one whit better off in Britain after the *Statute* than they were before. Nor were they in the nineteenth century.<sup>14</sup>

While this was the case in England, in France and Germany a different scenario was developing. The copyright regime in these countries was based more on personal rights with the author as the central figure. In these countries the author's personality was seen as embedded in her work. This inevitable bond between the author and the work was not severed when the work is sold to the publisher. Since the work was a reflection, and indeed, a part of herself, the author retains the right to defend it and, by extension herself, against alteration.<sup>15</sup> To this end moral rights were established, giving authors some power over their work. There are four generally accepted inalienable moral rights. They are; a) the right of disclosure, b) the right of attribution, c) the right of integrity, and d) the right of retraction.<sup>16</sup> This meant that even though the publishers could control the economic rights of the work, the moral rights rest with the author and cannot be given away.<sup>17</sup> Neither regime however, offers any special insight as to how to deal with the issues raised by the protection of software via copyright, or the creation monopolies in what are effectively functional works with, more often than not, a large number of authors. The main issue being the ease with which a monopoly can be created when copyright is applied to functional works.

Copyright was largely developed in the print era but the link between copyright and the creation of monopolies has not gone unnoticed. The Economic Council of Canada addressed the issue in their report on intellectual property with reference to the protection of software in modern times. The Council's report states:

The British law of copyright, of which the Canadian law is a lineal descendant, began as a child of print technology and State censorship. By one of the more fortunate ironies of history, it eventually so cut itself off from its second ancestor as to become a system of incentives to idea-processing involving singularly little day-to-day intervention by the State and therefore minimal opportunities and temptations to censor. In spite of much ingenious adaptation over the centuries, however, its link to its first ancestor

---

<sup>14</sup>*Ibid.*

<sup>15</sup>Jane C. Ginsburg, "A Tale of Two Copyrights: Literary property in Revolutionary France and America" (1990) 64 Tul. L. Rev. 991.

<sup>16</sup>Edward Damich, "The Right of Personality: A Common-Law Basis for the Protection of the Moral Rights of Authors" (1997) 23 Ga. L. Rev. 1.

<sup>17</sup>In Canada moral rights are recognized but may be legally waived.

remains strong, and this is at the root of many of the copyright issues faced today, when the once-predominant print technology, though still very much alive, competes with a rapidly growing variety of new means for the processing and transmission of information. One major problem, indeed, is how to relax somewhat the constricting tie of copyright to its first ancestor without bringing about a reincarnation of the second. That is, great care must be exercised to ensure that the necessary evolution of the incentive system, in parallel with technology, does not give rise to dangerous new possibilities of censorship of knowledge monopoly, whether on the part of the State or of private interests.<sup>18</sup> (Emphasis added.)

As the report asserts, the link between copyright and monopolies remains. The use of copyright to create monopolies manifests itself in much the same way as it did during the time of the Stationer's Company. Today, we have a limited number of transnational corporations who dominate the software industry. These corporations use copyright much like the printers did: to stifle competition, restrict the rights of authors, control access to information, as well as, controlling access to alternative innovative products. US District Judge Thomas Penfield Jackson, in the Microsoft anti-trust case, found the following: 1) Microsoft wields monopoly power in the personal computer industry; 2) Microsoft hurt consumers by stifling innovation, charging higher prices and selling a product that was susceptible to crashing and 3) Microsoft pressured other companies including Intel, Apple, RealNetworks and IBM to stop development of products that threaten its software.<sup>19</sup> Although this is an anti-trust (competition) case it is important to remember that the protection afforded to Microsoft's source code via copyright is one of the reasons that Microsoft became a monopoly in the first place. In fact, one of the remedies proposed to the Microsoft monopoly is to force the sharing of its Windows source code.

Furthermore, the inclusion of software developers under the copyright umbrella may dilute authors' rights in addition to users' rights. This is due to the fact that corporations or publishers own most software copyrights. According to Ralph Oman there is concrete ground for such concern. In discussing the WIPO treatment of software he states:

Of course, the traditionalists had some sound reasons for wanting to limit access to the exclusive copyright club, not just for reasons of aesthetics. They saw dangers to traditional authors and composers and artists by inviting these high-tech creators to join the party. Over the years, intellectual property has drawn its strength and its legitimacy from the great reverence we accord to works of the human mind. The WIPO had relied on this generally shared respect for the genius and hard work of our creators, and on the desire of governments to protect them, regardless of nationality, out of a sense of simple justice and fairness. Under the pressure of cultural nationalists and trade ministers who

---

<sup>18</sup>Economic Council of Canada, "Report on Intellectual and Industrial Property", January 1971.

<sup>19</sup>The Washington Post, "Judge Says Microsoft Wields Monopoly Power Over Rivals" November 6, 1999.

want revenues to go just one way—into the country—this old system of shared respect on which the WIPO had long relied began to crumble. For this reason, the copyright traditionalists worried that governments—willing to support high levels of protection for songs, plays, novels, and poetry—would be reluctant to give such generous support to high-visibility multi-billion dollar commercial products like computer software, databases, and sound recordings. Not only would the governments be reluctant to raise the level of protection if these commercial blockbusters were included, but they might even propose lowering protection across the board.<sup>20</sup>

Concern often arises when discussing ownership of copyrights in the software business, and the extent to which, not only copyright, but other available methods protect software. Given that software are created by people in the course of their employment at one corporation or another, copyrights in works created under such circumstances rest with the employer.<sup>21</sup> The need for software in today's society cannot be overstated. Virtually every tool we use in our everyday lives, works because of software. The concentration of a legal right as powerful as copyright in the hands of very few corporations (which employ thousands of creators) is cause for concern. This concern is further reinforced by the unsettled nature of copyright protection of software, creating a risk that there will be a substantial number of cases where the end result will be overprotection.<sup>22</sup> As discussed earlier, when copyright protection is

---

<sup>20</sup>Ralph Oman, "Canadian Copyright Revision: Does it Square with Canada's International Obligations?" Copyright Reform (Toronto: 1996)

<sup>21</sup>The *Copyright Act*, *supra*, note 4, s.13(3):

Where the author of a work was in the employment of some other person under contract of service or apprenticeship and the work was made in the course of the employment by that person, the person by whom the author was employed shall, in the absence of any agreement to the contrary, be the first owner of the copyright...

<sup>22</sup>Marci Hamilton and Ted Sabety, "Computer Science Concepts in Copyright Cases: The Path to a Coherent Law" (1997) 10 Harv. J.L. & Tech. 239

The controversy and case law regarding copyright protection for computer software has not yet exploited the concepts and precise definitions that are part and parcel of computer science to accurately define the extent of software copyright protection. This lack of rigor introduces anomalies into copyright law. As a first step in integrating computer science concepts and terminology into the discourse, this paper first demonstrates that if a court finds a data structure to be copyrightable expression because the structure, sequence, and organization of a program have been held protectable, it might confer monopoly power over those algorithms dependant on the given data structure. Second, if a court extends copyright protection to expressions of computer language grammar, it will confer monopoly power over all use of a given computer language—including expression not yet fixed. The first result violates the copyright statute because the statute excludes algorithms from copyright protection. The second result also violates the statute and frustrates the public policy of permitting migration paths between competing computer software products. Therefore, similarity between data structures (as distinguished from the data itself) and the similarity between expressions describing a computer language grammar that are necessary to construct a parsing program should be deemed non-infringement.

applied broadly, the end result is a monopoly. Therefore, copyright protection, which at times is applied too liberally and nearly always rests in the hands of a few corporations, leads to the situation we currently have: a select few corporations virtually own the software industry.<sup>23</sup>

The current antitrust suit against Microsoft illustrates how copyright in the hands of huge software businesses leads to monopoly issues. Howard Knopf discusses this problem in his article called "Intellectual Property Meets Trustbusters", he writes:

If Gates wins on the basis of intellectual property there will likely be nothing stopping him or anyone else who can invoke the exercise of an intellectual property right for any plausible business reason.... Copyright laws, which are vital to the computer industry, were originally intended to encourage and reward progress in the arts. Unlike patent law, with its industrial focus, copyright law historically attracted relatively little antitrust concern because artists cannot normally influence, let alone dominate an economic market.<sup>24</sup>

Software corporations being what they are, have the power to use copyright in ways which would bring about what existed in the past. Copyrights in the hands of the publishers affords them the power to limit public access to information. Most recently the software industry has helped to develop a new body of law called the *Uniform Computer Information Transactions Act* (UCITA) under the auspices of the *Uniform Commercial Code* (UCC) Article 2b to be passed by the various States. This piece of legislation has been characterized as one of the most anti-consumer legislation ever formulated.<sup>25</sup> A July 9, 1999 analysis by the Federal Trade Commission points out that UCITA allows software companies to place "restrictions on a consumer's right to sue for a product defect, to use the product, or even to publicly discuss or criticize the product." The analysis concludes, "we question whether it is appropriate to depart from these consumer protection and competition policy principles in a state commercial law statute."<sup>26</sup>

Monopolies in the software industry will continue since the authors contemplated throughout copyright's history has been seen as creators of creative original works, which convey information or have aesthetic value. They are not creators of functional works or tools. This is a very important distinction because of the different roles that aesthetic and functional works play in society. Obviously there are serious implications

---

<sup>23</sup>John G. Mills, "Possible Defences to Complaints for Copyright Infringement and Reverse Engineering of Computer Software: Implications for Antitrust and I.P. Law" (1998) 80 J. Pat. & Trademark Off. Soc'y 101.

<sup>24</sup>Howard Knopf, "Intellectual Property Meets Trustbusters", The Financial Post, July 4-6 1998.

<sup>25</sup>Mark Minasi, *The Software Conspiracy*, (McGraw-Hill, New York: 2000).

<sup>26</sup>*Ibid.*



inherent in the above analysis when deciding the appropriate legal protection for software. It is very instructive then, when deliberating legal protection for software that we to look at the features of software.

## Software

In the early years of the computer's evolution software played a minor role.<sup>27</sup> The established patent and trade secret law adequately provided much of the necessary intellectual property protection. These days, however, development efforts are much more focussed on software. The software industry has been greatly helped by extensive co-operation from within in those formative years.<sup>28</sup> The industry correctly decided that creating and adhering to standard protocols decreased the total investment required of each individual developer and allowed interoperability by users of various products, thus greatly increasing their market.<sup>29</sup>

Inherent in this co-operation was the ability to freely copy interfaces, file formats, protocols and even source code.<sup>30</sup> This liberal exchange of research was, for the most part, the catalyst for the exponential growth and development of the industry. This rapid growth has allowed the software industry to become one of the leading global industries.<sup>31</sup> The necessity for sharing in the software industry is illustrated in the following passage:

Computer science, though, differs fundamentally from all other sciences. Computer science has only one means of enabling peers to replicate results: share the source code. To demonstrate the validity of a program to someone, you must provide them with the means to compile and run the program.<sup>32</sup>

The principle purpose of software is to instruct the computer's hardware to carry out instructions. By serializing instructions to the hardware, software can achieve, complex operations consisting of millions of individual operations. Today's computers can carry out 100 million instructions in one second (MIPS). The greater the MIPS, the more complex functions the computer can effectively control. Some computers, such as those responsible for routing telephone calls, can carry out tens of billions of

---

<sup>27</sup>Mitchell Zimmerman, "Copyright in the Digital Electronic Environment", (1998) 527 *PLI/PAT* 543.

<sup>28</sup>*Ibid.*

<sup>29</sup>*Ibid.*

<sup>30</sup>*Ibid.*

<sup>31</sup>*Ibid.*

<sup>32</sup>Chris DiBona et al. *Open Sources*, (California: 1999)

operations in one second.<sup>33</sup> Even less complicated software applications require teams of engineers to assemble, since individual human beings can only understand the operation of a small part of the complete product. Each small part, called a module, is responsible for a single function. Each module can operate independently, as well as, a part of the whole. For the most part, software is written in source code form. Source code is a high level programming language in which people can easily express instructions to the computer. The source code is translated into object code, which the machine can understand.<sup>34</sup>

The computer needs to be able to readily copy the software directly into its memory so that if an operator wants to run the program she can do so quickly. Thus, the very features that make software useful to users equally make software easy to copy. As this implies, the software can be readily copied because the computer's operation requires it to be copied.

As we will see copyright is ill-equipped to deal with problems inherent in the world of software. The need for interoperability, the necessary borrowing between programmers and the utilitarian nature of software, help contribute to the skepticism that surrounds copyright protection of software. There is no doubt the copyright has had to accommodate many technologies since the Gutenberg press. However, software technology is not necessarily suited for copyright and vice-versa. As Sookman states:

Copyright has been stretched to the breaking point year after year in an effort to keep pace with technological developments. As applied to protection for computer programs, this has often resulted in decisions which reflect the court's attempt to fit the proverbial square peg in a round hole.<sup>35</sup>

This sentiment was reiterated in *Gates Rubber Co. v. Bando American, Inc.*<sup>36</sup> where the judge stated:

The issues in this case stem from the traditional conflict in copyright law—how to protect an author's creative expression—while preserving competition in the marketplace. This dilemma is nothing new, but in the case law and commentators in the area of copyright protection seem woefully ill-equipped to provide a systematic means for analyzing copyright issues as they arise in the context of computer software. Indeed, the heart of copyright law, designed to accommodate unimaginable varieties of creative expression, has mandated resolution of disputes on a case-by-case basis. What magnifies

---

<sup>33</sup>*Ibid.*

<sup>34</sup>*Ibid.*

<sup>35</sup>Barry Sookman, *Computer Law: Acquiring and Protecting Information Technology*, (Toronto: 1997).

<sup>36</sup>*Gates Rubber Co. v. Bando American Inc.*, (1992) 24 USPQ 2d 1161.

the underlying dilemma however, is the realization that copyright law was not designed to accommodate computer software protection.<sup>37</sup>

It is easy to see why copyright is such an inappropriate method of legal protection if one assumes the generally accepted goals of copyright to be; public access to information, the protection of the rights of users and authors and progress through free flowing ideas. As mentioned earlier one of the best ways to improve the quality of software is to make the source code publicly available instead of protecting it via intellectual property laws. Unlike other works covered under the copyright regime, the societal interest in ensuring that we have top quality software is great. This is because of the functional nature of software. Software make devices work and when software do not work properly devices fail. Aside from the thousands of hours of work lost due to computer bugs and the number of hours spend trying to receive technical help for bugs known to the industry, there is an even more crucial reason to insist on quality software.

Software do not just run home computers: they are also responsible for the functioning of cars, airplanes, safety equipment, wapons, power plants etc. and when these software have bugs, the results can be deadly. For example, in *General Motors v. Johnson*<sup>38</sup> the court ordered GM to pay \$7.5 million to the plaintiff because a software bug, known to GM, caused the plaintiff's car to stall at an intersection where an oncoming truck hit the car killing the plaintiff's grandson. It was revealed that GM did not feel the need to fix the software bug because it was not a structural defect.<sup>39</sup> Also, Flight 801 would not have crashed in Guam, killing 224 people if a piece of software called Radar Minimum Safe Altitude Warning system was better written. In this case, the software, which tells airport controllers if a plane is flying too low, caused a lot of false alarms and instead of writing better code for the program the company elected to desensitize it.<sup>40</sup> This means that the sensor instead of sensing low-flying planes for 63 miles, it would only sense them for 1 mile. There are numerous other cases in which badly written software has caused harm, such as hospital radioation control software which limit the amount of radiation used to take x-rays, but computer bugs seem to be treated differently then defects in other products.<sup>41</sup>

The reason for this is that there seems to be a perception that low-bug software is an impossibility, but this is not the case. One of the most stable operating systems is Linux. Linux, source code and all, can be downloaded from the Internet for free. How

---

<sup>37</sup>*Ibid.*

<sup>38</sup>592 So. 2d 1054, 1992 (Supreme Court of Alabama).

<sup>39</sup>Minasi, *supra* at note 25.

<sup>40</sup>*Ibid.*

<sup>41</sup>*Ibid.*

is it that you can have free, stable software? Well, Linux is developed under the Open Source concept. Software developers, in this case, believe that the best way to ensure quality software is to share the source code, thus allowing programmers from all over the world to contribute to its betterment. According to Cella, the Open Source strategy is "to ensure that code released into the community remains open and is not subsequently removed from the programming community by developers who modify the code and release the modified code under traditional source code and copyright traditions."<sup>42</sup> The Open Source model was adopted by Netscape in 1998. Within hours of making the code available, the company received code for a security patch from an Australian group of programmers and numerous other improvements came in from all over the world. In less than a month Netscape had a new version of its browser on the market. This was impressive enough to catch the attention of Sun Microsystems and Apple Computers both of which are now releasing some of their products with the source code.<sup>43</sup> Interestingly enough, however, about the only thing certain when applying copyright law to software is that the source code is protected.

### Copyright and Software

There are certain principles that have evolved over time to become part of the copyright law tradition. The Canadian copyright essentials were outlined by the court in *Delrina Corp. v. Triolet Systems Inc.*<sup>44</sup> as stated in *Delrina* copyright protects only

---

<sup>42</sup>Charles Cella, "Considerations for companies developing software under the Open Source model" (1999) 4 *Cyberspace Law*. 9.

<sup>43</sup>*Ibid.*

<sup>44</sup>*Delrina Corp. v. Triolet Systems Inc. et al* (1993), 47 C.P.R. (3d) 1 (Ont. Gen. Div.) It is useful to enumerate some general principles applicable to the law of copyright.

- An author has no copyright in ideas or information, but only in his expression of them.
2. Copyright subsists in original literary works. There is no copyright in what the author has copied from something already in the public domain or from a work in which another holds the copyright.
  3. Even if the expression originated with the author, the expression of the idea is not copyrightable if the expression does no more than embody elements of the idea that are functional in the utilitarian sense.
  4. If an idea can be expressed in only one or in a very limited number of ways, then copyright of that expression will be refused for it would give the originator of the idea a virtual monopoly on the idea. In such a case it is said that the expression merges with the idea and thus is not copyrightable.
  5. Copyright does not subsist "in any arrangement, system, scheme, method for doing a particular thing, procedure, process, concept, principle, or discovery, but only in an author's original expression of them". Consistent with accepted thinking in copyright law, therefore, a particular expression of a mathematical algorithm or other procedure for solving a problem or accomplishing some end in the form of sets of instructions or statements may be protected by copyright, but the mathematical algorithm or other

the expression of an idea, not the idea itself<sup>45</sup> and copyright does not ordinarily apply to functional creations.<sup>46</sup> Under current legislation computer programs and compilations are protected as literary works, regardless of the medium in which such programs are expressed.<sup>47</sup> Copyright protection attaches to any copyrightable item from the moment it is created and fixed in a tangible form.<sup>48</sup> Therefore, computer programs are granted full copyright protection as soon as they are created.<sup>49</sup>

In order for a work to qualify for copyright protection it must be original. In *International Business Machines Corp. v. Ordinateurs Spirales Inc./Spirales Computers Inc.*<sup>50</sup> the court found that a computer program in its source code form has been found to meet the necessary criteria of representing an "... expression of thought in an original form". The word "original" however, does not necessarily imply novelty. The level of originality does not have to be very high. According to Nimmer:

Originality means only that the work owes its origin to the author, i.e., is independently created, and not copied from other works. Therefore a work is original and may

---

procedure as such cannot be protected by copyright.

<sup>45</sup>Also see: *Matrox Electronic Systems v. Gaudreau* [1993] R.J.Q. 2449 (Que. Sup. Ct.): "It is fundamental that copyright can protect the form of expression of computer programs, but not the ideas embodied therein."

<sup>46</sup>Also see: *Lotus v. Paperback*, 740 F. Supp. 37 (D. Mass. 1990) at pp. 57-8.

<sup>47</sup>*Copyright Act, supra* at note 4.

<sup>48</sup>George S. Takach, *Computer Law*, (Toronto: Irwin Law, 1998). There are further technical issues associated with using copyright to protect software beyond the scope of this paper such as the debate over RAM memory: According to Kristen J. Mathews in her article, "Misunderstanding RAM: Digital Embodiments and Copyright" (1997) B.C. Intell. Prop. & Tech. F. 04150:

Furthermore, including digital works embodied in RAM as reproductions is a poor fit in light of the policy behind the Copyright Act. This would mean that every time a person opens a computer program, he or she might be infringing a copyright. The courts' widely criticized finding can be explained, at least in part, by early law makers' confusion about computer memory and inability to fit RAM into previous constructs. Courts and law makers have built on each others' flawed or non-existent analysis of RAM embodiments as reproductions since the 1976 Act was being drafted. These approaches ignored the purpose behind the fixation requirement when interpreting it. Since digital embodiments in RAM do little harm to a copyright holder, their categorization as reproductions is not consistent with the policy behind the fixation requirement for reproductions.

<sup>49</sup>After the 1988 amendments to the *Copyright Act*, the criminal punishment for copyright infringement was greatly increased. A summary conviction carries a 6 month jail term or a \$25,000 fine and an indictable offence 5 years in jail or \$1,000,000 fine. There is reason to believe that the increase in penalties is a direct result of an effort to accommodate the needs of software and database developers. Gordon F. Henderson, *Copyright and Confidential Information Law of Canada* (Toronto: Carswell: 1994) at 22.

<sup>50</sup>(1984), 80 C.P.R. (2d) 187.

command copyright protection even if it is completely identical with a prior work provided it was not copied from such prior work but is rather a product of the independent efforts of its author.<sup>51</sup>

Computer software, for the most part, meets the originality requirement. However, copyright generally protects original *non-functional* works.<sup>52</sup> George Takach explains in his book *Computer Law*:

The *Copyright Act* has always been an uncomfortable home for software. Affording copyright protection to computer programs by calling them literary works has always been an effective and efficient way of combating wholesale piracy, the practice of reproducing all or almost all of a computer program and selling the illegal copy on a bootleg basis. By amending the definition of literary work in the *Copyright Act* to cover computer programs in 1988, software developers were given quick protection in Canada and abroad through the *Berne Convention*. The alternative of crafting a separate legal regime for software, as has been done with chip topography technology, would have resulted in a much slower pace of protection both domestically and globally. It is, nonetheless, something of a fiction to call software a literary work. Novels, plays, art and music, the traditional core copyright works, are communicative vehicles intended to express artistic or aesthetic values. The real genius in these types of works is their expressive flair. Of course Shakespeare crafted intriguing plots and created notorious characters, but his really profound contribution to English literature is his dialogue, the actual words he chose to express and give life to his eternal themes. There is no correct or best way to write about a love between two young people whose families stand in the way. Shakespeare did it one way in *Romeo and Juliet*, but Leonard Bernstein expressed it in another way in *West Side Story*. Monet and Cezanne both painted the French countryside, but with much different styles—each with his own expressive print; the same can be said of Emily Carr and Tom Thomson with respect to Canadian landscape painting. In contrast, a computer program that runs a company's payroll is a utilitarian device that controls a machine to perform certain predetermined functions. Other software process documents, sorts data, performs calculations; these are very different activities than the purpose of a book, which is simply to convey information. Even maps and charts, which have long been covered by copyright, merely convey information—they do not operate machines.<sup>53</sup>

As suggested by Takach, software in a sense falls under the category of literary works. The source code is written in letters and the object code does consist of 1's and 0's, but it is easy to see how the literary character awarded to software can be

---

<sup>51</sup> Nimmer, *The Law of Copyright*, (1982).

<sup>52</sup> Robert L. Bocchino Jr. "Computers, Copyright, and Functionality: The First Circuit's Decision in *Lotus Development Corp. v. Borland International, Inc.*" (1996) 9 Harv. J.L. & Tech. 467.

<sup>53</sup> Takach *supra*, at note 48.

questionable. In *Dynabec Ltee. v. La Soci t  d'Informatique R.D.G. Inc.*<sup>54</sup> the court reproduced part of a source code:

```
Ok
LIST
10 LPRINT<La Compagnie d=Information ABC>:
  LPRINT:LPRINT:LPRINT:LPRINT
20 LPRINT TAB (31) <MON CLIENT LTEE>
30 LPRINT:LPRINT:LPRINT
40 LPRINT TAB (29) <LISTE DU SALAIRE BRUT>:
  LPRINT:LPRINT:LPRINT
50 LPRINT TAB (10) <NOM DE L=EMPLOYE>,
  TAB (30) <TAUX>, TAB (45)<HEURES>, TAB (60)
  <BRUT>
60 LPRINT:LPRINT:LPRINT
70 CLS LPRINT<La Compagnie d=Informatique
  ABC>: LOCATE 12,25
80 INPUT<Entrez le nom de l=employe>;
  NOM: LOCATE 12,25
90 INPUT<Entrez son taux horaire>;
  TAUX :LOCATE 14,25
100 INPUT<Entrez le nombre d=heures>;
  HEURES...55
```

The implication, from the above quotations, is that copyright was a default choice for protecting software and not because copyright is an especially appropriate vehicle for protecting this area. Karjala writes:

Why did we suddenly turn to copyright law for the protection of such intrinsically functional works? Calling programs "literary works" is simply another way of phrasing the question. We could equally, perhaps even more aptly, call them "methods of machine design," in that they take a universal machine and transform it into one that achieves a particular result. The real reason we resorted to a copyright scheme to protect computer programs is that many programs-including programs that are costly and time consuming to develop-are simply the result of technologically straightforward application of well-known programming principles to a well-defined problem. These programs do not meet the requirement of traditional patent law for a nonobvious advance in the art. Yet, once these programs are distributed in object-code form, they can be copied almost without cost in large numbers. Without some form of protection, we should expect that they would be underproduced. Because the evil to be avoided was thus slavish copying, especially slavish electronic copying, because copyright protects at least against that, and because computer programs formally fit the broad definition

---

<sup>54</sup>(1985), 6 C.P.R. (3d).

<sup>55</sup>*Delrina, supra* at note 42.

of a literary work under copyright law, it became a natural candidate for the protection of programs, notwithstanding their inherent functionality.<sup>56</sup>

Focussing on the utilitarian nature of software does not imply that there is no creativity involved in producing these products. Both patents and copyright protect creativity, but copyright protects non-functional creative works. The problem with software is that although they are functional works, until very recently, even the most complex of programs did not qualify for protection under the patent regime. Taking this into consideration and the fact that computer software developers need expedient protection, as opposed to the years it takes to get through the patent process, "copyright was a convenient at-hand tool for achieving the desired result."<sup>57</sup>

Adding to the confusion as to the nature of functional and non-functional works is the tendency to equate functional with useful. This is problematic as it blurs the distinction between the subject matter available for patents and copyright. Functionality, properly defined, is one way to understand why two statutory mechanisms were created to protect intellectual property. Copyright protects a number of "useful" items, such as, maps, recipes etc.<sup>58</sup> However, there are not useful in the sense that useful is applied in the *Copyright Act*. In the American *Copyright Act* useful articles are defined as "an article having intrinsic utilitarian function that is not merely to portray an appearance of the article or to convey information".<sup>59</sup>

This utilitarian nature of software and the deleterious effects of protecting software were extensively discussed in *Lotus Development Corp*. The First Circuit court stated:

Most of the law of copyright and the "tools" of analysis have developed in the context of literary works such as novels, plays, and films. In this milieu, the principal problem—simply stated, if difficult to resolve—is to stimulate creative expression without unduly limiting access by others to the broader themes and concepts deployed by the author. The middle of the spectrum presents close cases; but a "mistake" in providing too much protection involves a small cost; subsequent authors treating the same themes must take a few more steps away from the original expression.

The problem presented by computer programs is fundamentally different in one respect. The computer program is a means for causing something to happen; it has a mechanical utility, an instrumental role, in accomplishing the world's work. Granting protection, in other words, can have some of the consequences of patent protection in

---

<sup>56</sup>Dennis S. Karjala, "A Coherent Theory for the Protection Computer Software and Recent Judicial Interpretations" (1997) 66 U. Cin. L. Rev. 53.

<sup>57</sup>*Ibid.*

<sup>58</sup>Bruce Vogel, "Copyright Protection of Software and Compilations a Review of Critical Developments 1991-1997" (1997) 481 PLI/Pat 157.

<sup>59</sup>Karjala, *supra*, at note 56.



limiting other people's ability to perform a task in the most efficient manner. Utility does not bar copyright (dictionaries may be copyrighted), but it alters the calculus.

Of course, the argument for protection is undiminished, perhaps even enhanced, by utility: if we want more of an intellectual product, a temporary monopoly for the creator provided incentives for others to create other, different items in this class. But the "cost" side of the equation may be different where one places a very high value on public access to a useful innovation that may be the most efficient means of performing a given task. Thus, the argument for extending protection may be the same; but the stakes on the other side are much higher. It is not accident that the patent protection had preconditions that copyright protection does not—notably, the requirements of novelty and non-obviousness—and that patents are granted for a shorter terms than copyrights. This problems of utility has sometimes manifested itself in copyright cases, such as *Baker v. Selden*, 101 US 99 (1879), and been dealt with through various formulations to limit copyright or create limited rights to copy. But the case law and doctrine addressed to utility in copyright have been brief detours in the general march of copyright law....

Thus, to assume that computer programs are just one more new means of expression, like a filmed play, may be quite wrong. The "form"—the written source or the menu structure depicted on the screen—look hauntingly like the familiar stuff of copyright; but the "substance" probably has more to do with problems presented in patent law or, as already noted, in those rare cases where copyright law has confronted industrially useful expressions. Applying copyright law to software programs is like jigsaw puzzles whose pieces do not quite fit.

As stated in the above quotation the cost associated with overprotection of tools, necessary to perform daily tasks, is far greater that that of purely informational works. If we look at case law concerning software protection we see that although initially copyright appeared to be good way to provide protection for software, the courts soon realized the consequences of such protection and tried to narrow that protection afforded.

*Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.*<sup>60</sup> is one of the first cases to deal with the problem of what copyright protects in software. The court gave such broad copyright protection to software that structure, sequence, organization and non-literal elements of a program were deemed protected by copyright.<sup>61</sup> In Canada, the

---

<sup>60</sup>797 F.2d 1222 (3d Cir. 1986).

<sup>61</sup>It is a well established fact that literal elements of a computer program, the source and object codes, are protected by copyright.

broadest protection awarded to software under copyright law was in the *Gemologists*<sup>62</sup> case decided just after *Whelan*.<sup>63</sup>

The cases of *E.F. Johnson Co. v. Uniden Corp. of America Inc.*<sup>64</sup> and *Lotus Dev. v. Paperback Software*<sup>65</sup> attempted to revert to a slightly narrower interpretation of what is copyrightable in light of *Whelan*. It was in *Autoskill Inc. v. National Educational Support Systems Inc.*<sup>66</sup> however, that the judge refused to follow the “look and feel cases” and said:

A better approach for determining what is idea as opposed to expression is known as the abstractions test articulated by Judge Learned Hand in *Nicholls v. Universal Pictures Corporation* (17 USPQ 84 2nd Cir 1930)... [u]pon any work, and especially a play, great number of patterns of increasing generality will fit equally well, as more and more of the incident is left out. The last may be no more than the most general statement of what the play is about, and at times may consist of only its title; but there is a point in this series of abstractions where they are no longer protected, since otherwise the playwright could prevent the use of his ideas to which, apart from his expression, his property never extended.<sup>67</sup>

This idea was more precisely developed in *Computer Assoc., Int'l., Inc. v. Altai, Inc.*<sup>68</sup> The court developed a 3-part test which is known as the abstraction-filtration-comparison test. This is basically a part test in which the court analyzes the levels of abstraction in a piece of software, strip away all the levels that are not protectable and what is left is the protectable aspect of the software.<sup>69</sup>

In Canada, *Delrina* and *Matrox* considerably narrowed the scope of protection. In *Delrina*, after a very lengthy review of US cases the court decided:

Whether a Canadian court should adopt the abstraction-filtration-comparison method in deciding an action for copyright infringement or some other similar method, it seems clear that before a computer program or some part of it can be held to be copyrightable, some method must be found to weed out or remove from copyright protection those portions which, for the various reasons already mentioned, cannot be protected by

---

<sup>62</sup>*Gemologists International Inc. v. Gem Scan International Inc.* (1986), 7 C.I.P.R. 255.

<sup>63</sup>*Ibid.*

<sup>64</sup>26 USPQ 2d 1828 (1993).

<sup>65</sup>740 F. Supp. 37 (D. Mass. 1990).

<sup>66</sup> 24 U.S.P.Q. 2d 1107 (D.C. New Mex. 1992)

<sup>67</sup>*Ibid.*

<sup>68</sup>982 F.2d 693 (2d Cir 1992).

<sup>69</sup>For a full discussion on this test see: Michael Morgan, “Trash Talking: the Protection of Intellectual Property Rights in Computer Software” (1994) 26 Ottawa L. Rev. 425.

copyright. After the portions that are not copyrightable have been filtered out, there may or may not be any kernels or golden nuggets left to which copyright can attach.<sup>70</sup>

The functional nature of software, together with the difficulty in separating ideas from expression, are the main characteristics which makes software a bad candidate for copyright protection. This is obviously very dangerous considering the importance of the free flow of ideas. The free flow of ideas is important for the creation of innovative products, as well as, the progress of society in general. The protection for functional works, which generally tend to incorporate the idea into the creation, is governed by a substantially different set of rules. Furthermore, software publishers have technological means by which to protect their products in addition to various intellectual property measures.

### Copyright, Software and Technological Barriers to Infringement

The Business Software Alliance (BSA) estimates that the software industry lost more than \$13 billion in 1995 to global piracy.<sup>71</sup> Although digital technology may advance the cause of software piracy, it may also be used to fight it. Historically, technology has meant copy protection, and copy protection is met with skepticism because of past failures.<sup>72</sup> In the mid-1980s, software publishers turned to copy protection in an attempt to physically prevent pirates from making unauthorised copies of software.<sup>73</sup> According to Kory Christensen they worked in the following manner:

For instance, each time a user wished to operate a copy-protected program, he or she had to insert an original "key disk" into the disk drive. If the key disk was lost or damaged, the computer was unable to operate the software. The industry soon turned away from the practice of copy protection for two reasons. First, publishers learned that every copy protection scheme, no matter how sophisticated, was eventually "cracked" (or defeated) by an equally clever hacker. Some analysts opined that the lifetime of any given copy protection scheme was between three and four months. Second consumer preference led to the abandonment of copy protection because "users thought [it] interfered with legitimate uses of the software." Consumers were unhappy because they could not make legitimate backup copies of their expensive programs. If the key disk became lost or damaged, software owners had to wait for the manufacturer to send a

---

<sup>70</sup>*Delrina, supra* at note 44.

<sup>71</sup>[www.bsa.org](http://www.bsa.org)

<sup>72</sup>Thomas C. Vinje, "A Brave New World of Technical Protection Systems: Will There Still Be Room For Copyright?" (1996) 8 EIPR 431.

<sup>73</sup>*Ibid.*

replacement. Moreover, users objected to the inconvenience of having to insert key disks whenever they used the software.<sup>74</sup>

Cryptography could potentially solve many of these earlier problems with security systems. It has long been used by military and intelligence agencies as a method to protect the integrity of communications.<sup>75</sup> Encryption involves a process which renders text unintelligible to anyone without the key.<sup>76</sup> Both encryption and decryption are accomplished by means of complex mathematical algorithms.<sup>77</sup> Modern algorithms use keys which are strings of alphanumeric digits to encrypt and decrypt messages.<sup>78</sup> The length of the key determines the strength of the encryption: long keys can produce virtually unbreakable security. For example, to decrypt a 128-bit key would require a computer capable of processing one million keys per second over  $10^{25}$  years to break the code.<sup>79</sup>

With the development of faster computers, the science of cryptography can now be applied to many new applications. One such application is encrypting computer programs to protect them from piracy.<sup>80</sup> Cryptography for computer programs is not a complicated procedure. Computer programs are similar to text messages so the process is almost the same. Christensen describes them in the following way:

Programs are represented by "bytes," or characters, and are stored in the same memory space as other documents. Therefore, like secret messages, computer programs can be encrypted with unique keys before they are distributed to the public. An encrypted software package would also include a decryption routine capable of unscrambling the software when the user provides the appropriate key. Such keys would generally accompany commercial software and would be registered to the purchaser. Alternatively, purchasers may be required to contact the manufacturer by telephone or Internet to obtain the key. For even greater security, keys could be encapsulated in hardware.<sup>81</sup>

---

<sup>74</sup>Kory Christensen, "Fighting Software Piracy in Cyberspace," (1997) 28 *Law & Pol'y Int'l Bus.* 435.

<sup>75</sup>*Ibid.*

<sup>76</sup>Information Infrastructure Task Force, *Report of the Working Group on Intellectual Property Rights, Intellectual Property and the National Information Infrastructure*, 66 (1995).

<sup>77</sup>*Ibid.*

<sup>78</sup>Thomas Smedinghoff, *Online Law* (New York: 1996).

<sup>79</sup>*Ibid.*

<sup>80</sup>Michael Froomkin, "The Metaphor is the Key: Cryptography, the Clipper Chip, and the Constitution" (1995) 143 *U. Pa. L. Rev.* 709.

<sup>81</sup>Christensen, *supra* at note 74.

The advantages of cryptography are many; the user can make back up copies if she wishes, breaking in via hacking is far more difficult, it can easily be implemented over networks, etc. In fact in most cases the user need not know it is there since it does not affect any computer functions.<sup>82</sup>

The best part of this technology is that it is borderless. Changes in copyright legislation et cetera do not affect the level of protection. The only glitch to this scheme is that the US government has placed restrictions on certain types of encryption technology. However, these restrictions do not necessarily apply to software-protection systems. The law expressly exempts programs that are "restricted to decryption functions specifically designed to allow the execution of copy protected software, provided the decryption functions are not user-accessible."<sup>83</sup>

Innovations in encryption and other technologies designed to prevent unauthorized use of a product create a further imbalance toward the copyright owner as opposed to the rest of society. The imbalance grows with the realization that the technological means of protection are in turn protected from being tampered with by further legislation. In August 1998 the US Congress passed a bill called the *Digital Millennium Copyright Act*<sup>84</sup> specifically designed to protect software anti-piracy technology from being circumvented. There are stiff penalties for compromising technologies inserted into products by copyright holders. The bill was supposed to implement the WIPO copyright treaties but ended up being much more.<sup>85</sup>

The combination of copyright protection, technological protection and legislation against the tampering with anti-piracy devices, gives the software industry far too much protection. This industry should not have this protection given the utilitarian nature of its products and the role these products play in modern economy and society. One way to ameliorate the situation is to expand the fair dealing doctrine of the *Copyright Act*. The fair dealing provisions should expressly allow for the sharing of code, reverse-engineering and to recognize the role that software play in society and adequately protect users.

### Copyright, Software and Users

Users have several rights under copyright. These users' rights include: 1) the right to use a work for criticism, comment, news reporting, scholarship, or research; (fair

---

<sup>82</sup>Pamela Samuelson, "Technological Protection for Copyrighted Works" (1996) 45 Emory L.J. 217.

<sup>83</sup>See SPA website for more technological ways to stop piracy. [www.spa.org](http://www.spa.org).

<sup>84</sup>*Digital Millennium Copyright Act* (s.2037).

<sup>85</sup>[www.abcnews.com/sections/tech/CNET/cnet\\_copyright0515.html](http://www.abcnews.com/sections/tech/CNET/cnet_copyright0515.html)

dealing) 2) the right of libraries to make single copies for interlibrary loan programs and to provide photocopy machines for public use so long as a copyright notice is posted; 3) the right of an owner of a lawful copy of a copyrighted work to sell or otherwise dispose of it; 4) the right of teachers or students in non-profit educational institutions to perform or display works in instructional settings; and 5) the right of an owner of a lawful copy of a computer program to make a backup copy and adapt the program for use with a particular machine.

The dilemma between rewarding the creator while allowing the public access to the creation was dealt with in *Sayre v. Moore*<sup>86</sup> where the court stated:

The rule of decision in this case is a matter of great consequence to the country. In deciding it we must take care to guard against two extremes equally prejudicial; the one, that men of ability, who have employed their time for the service of the community, may not be deprived of their just merits, and the reward of their ingenuity and labour; the other, that the world may not be deprived of improvements, nor the progress of the arts be retarded.<sup>87</sup>

This concern was recently reiterated in the Preamble of the *1996 World Intellectual Property Organization (WIPO) Copyright Treaty*, where it states:

The Contracting Parties,...

Emphasising the outstanding significance of copyright protection as an incentive for literary and artistic creation,

Recognising the need to maintain a balance between the rights of authors and the larger public interest, particularly education, research and access to information, as reflected in the Berne Convention,

The maintenance of the balance between the interest of society and those of the copyright holders is crucial for an intellectual property-importing nation like Canada. Canada needs to seriously examine the effects on Canadian society of such strong protection of vital technology. The Economic Council report addressed this issue in the following manner:

Even if Canada greatly improves its performance as a knowledge producer and a purveyor of information internationally, its balance of international payments for information will likely be always heavily outbound, and this fact should be kept clearly in mind for purposes of international negotiation. The maintenance of good access to foreign information is crucially important for Canada and it should be the lowest cost

---

<sup>86</sup> (1785) 1, East's Report 361 (K.B.)

<sup>87</sup> *Ibid.*

access obtainable, consistent with Canadian consumers paying a fair share of reasonable incentive to authors and other copyright holders and assignees the world over.<sup>88</sup>

A country like Canada, needs to protect itself from being excluded from the information loop due to its citizen's inability to pay whatever information monopolies decide to charge. According to Judge Jackson in the Microsoft anti-trust case, Microsoft could have easily charged \$49 for a Windows 98 upgrade but decided instead to charge \$89 in order to maximize its profit.<sup>89</sup>

There is also a concern about monopolies created via intellectual property shared by people in the software industry, as it becomes more and more evident that the larger corporations are not allowing smaller ones to compete. The following results from a survey reported by Burton show this alarming trend:

By 79.6% to 8.2%, the computer programmers said that granting patents on computer software impedes, rather than promotes, software development. (The remaining 12.2% were undecided.) By 59.2% to 26.5%, most went even further, saying that software patents should be abolished outright. (Current U.S. law allows the patenting of computer software algorithms, but many other nations do not recognize such patents.) The programmers are even more strongly opposed to copyrights on the "look and feel" of software user interfaces. By 85.7% to 8.2%, they think that such copyrights impede, rather than promote, software development. By 77.6% to 14.3%, they want to abolish such copyrights.<sup>90</sup>

As we have seen, the link between copyright and monopolies is a strong one. This link is very disturbing when the scope of copyright is extended to include functional tools such as software. The role that software plays in our society is very different than that of traditional copyrightable works. Monopolies in the software industry affect a wide segment of the population. As it was said previously, the cost of overprotecting in industries such as software is tremendous. Lack of innovative software, the undermining of user rights, the dilution of authors' rights, are all results of the monopolies created with the help of misapplied intellectual property laws. As conceded earlier, copyright will most probably not be replaced as the legal method of protection for software but the scope of protection can be limited. As we have seen, the courts seem have taken this route, upon the realization that copyright provided too much protection for these works.

---

<sup>88</sup>Economic Council Report, *supra* note 18.

<sup>89</sup>The Washington Post, *supra* note 19.

<sup>90</sup>Burton, "Software Developers Want Changes in Patent and Copyright Law" (1996) 2 Mich. Tel. Tech. L. Rev. 2.

## Conclusion

There needs to be clear legal precedent provided as to the exact scope of copyright protection for software. As it stands now, there is much confusion about the what protection is afforded to software. As new technologies emerge this confusion will grow. For example how to protect works created by software, or multimedia works. It is vital that courts clarify this area of law. This is especially true for Canada, given our position as a software importer. If copyright is to fulfill its purpose as a method for the promotion of ideas and information; the protection of authors and users; and the discouraging of monopolies by disallowing ownership in ideas, a clear limitation to the scope of protection for software needs to be set. This can be done either within the existing copyright laws through the expansion of the fair dealing provisions or via a sui generis system. There is no doubt that software needs legal protection. In fact its vital role in society necessitates protection for such an important tool. The protection however should not lead to monopolies and the stifling of innovation by others.

In today's world it is virtually impossible to lead a productive life without the use of software to perform numerous utilitarian tasks and to access information. The tools which make all this possible need to be protected in a manner which takes into consideration not only the unique nature of these tools but also their role in society. As copyright was designed with very different works in mind, protecting software via copyright leads to overprotection. Furthermore, copyright in conjunction with technological barriers to infringement virtually ensures overprotection. It is important, especially for countries like Canada, to re-evaluate the protection afforded to software, so that public access is not restricted and innovation is encouraged.