Osgoode Hall Review of Law and Policy

Volume 3
Number 1 *Volume 3, Number 1 (March 2010)*

Article 4

2010

Contextualizing the Software Patent Debate in Canada: A Practical Approach to Policy Development

Conrad Delbert Seaman

OSGOODE HALL REVIEW OF LAW AND POLICY

VOLUME 3 MARCH 2010 ISSUE 1

ARTICLE

CONTEXTUALIZING THE SOFTWARE PATENT DEBATE IN CANADA: A PRACTICAL APPROACH TO POLICY DEVELOPMENT

Conrad Delbert Seaman*

There has been ongoing international debate regarding the patentability of software for at least 15 years. Despite being bound by international laws, which deal directly with the patentability of software (TRIPS), individual countries continue to justify vastly different legislative and practical patent regimes in this field of innovation. In a very traditional and conservative fashion Canada has placed itself carefully on the fence in this debate adopting an approach which falls somewhere between that of the U.S. and Europe, providing little practical guidance for businesses, lawyers or software developers in the industry. This paper seeks to establish an approach, solution and justification for the correction of these problems.

Discussion in the area of software patents is often based substantially around patent law theory and statistical analysis. Such approaches disregard the context in which these laws operate. As a direct consequence the connection between software patents and innovation remains an area of substantial conjecture. As the basis for policy decisions this non-contextual approach leaves much to be desired. In Canada this situation is amplified by

* Conrad Seaman is the technical lead and Senior Forum Editor for the *Osgoode Hall Review of Law and Policy.* He holds a B.Math in Computer Science from the University of Waterloo and a Juris Doctorate from Osgoode Hall Law School. Prior to law school he worked in the software development and advertising industries as a technical contractor and project manager for over five years. Conrad also studied philosophy in his undergrad and holds a religious studies minor. Conrad is articling with an IP boutique on Bay St. and intends to focus on technology transfer, patents and intellectual property licensing during his career.

the fact that few significant efforts have been made to study the legal effects of patents on the Canadian software sector.

Recognizing the gap between theory and practice this paper seeks to marry the academic debate over software patentability with concrete Canadian perspectives from inside the industry. To this end primary research based on personal interviews with representatives from three software companies, with innovation offices in Canada, is used to shed a contextual, Canadian and practical light on U.S. and EU patent law theory. The trend which emerged from these interviews was that Canadian software companies generally find software patents detrimental to their business objectives. Given the small sample size, confirmation of this trend within the broader Canadian software industry is not possible and further research is required in order to substantiate this papers recommendations. However, assuming that the software companies interviewed are representative of the Canadian software industry then the detrimental trend identified aligns the Canadian software industry with non-innovation theories of software patentability. This allows the paper to justifiably conclude that Canada should not extend patentability to software or in the alternative that a carefully considered extension of patent law which responds systemically to the unique needs of the software industry and other emerging technologies may be appropriate. Most importantly the paper stresses the practical importance of active contextual research during the development of clear and strong guidelines related to the patentability of software in Canada

TABLE OF CONTENTS

INTRODUCTION

Over time patent law has continually increased its range of influence and with each expansion debate over its merits and value has never been far behind.[1] In the software development sector there has been ongoing international debate regarding patentability for at least 15 years.[2] In most cases debate on such topics normally gives way to general acceptance as the value of patents becomes recognized in the newly enveloped sector.[3] This has arguably not occurred in the software industry. Despite being bound by international laws, which deal directly with the patentability of software, individual countries continue to justify vastly different legislative and practical approaches

[1] Edith Tilton Penrose, *The Economics of the International Patent System*, (Baltimore: Johns Hopkins Press, 1951). Penrose shows that industry in the 17th and 18th century vehemently opposed the patenting of mechanical devices.

[2] Cases in the U.S. including *Re: Alappat*, 33 F.3d 1562 (1994) and *State Street Bank v. Signature Financial Group,* 149 F.3d 1368 (1998) triggered significant debate as the U.S. committed to its approach on software patentability.

[3] *Supra* note 1. As an example the vehicle industry, initially opposed to patenting, became one of its principle advocates in a period of less than 10 years.

to the problem.[4] In a very traditional and conservative fashion Canada has placed itself carefully on the fence in this debate adopting an approach which falls somewhere between that of the U.S. and Europe. The result is best described as a non-position on software patentability providing little practical guidance for businesses, lawyers or innovators in the industry.

Discourse in the sphere of software patents is predominantly focused on the approaches of the U.S. and EU due in large part to both their size and economic influence as well as the divergence of their solutions. Furthermore, discussion is often based substantially around patent law theory and statistical analysis. Such an approach disregards the context in which these laws operate.[5] As a direct consequence the connection between software patents and innovation remains an area of substantial conjecture. As the basis for policy decisions this non-contextual approach leaves much to be desired. In Canada this situation is amplified as few significant efforts have been made to study the legal effects of patents on our software sector.[6]

Recognizing the gap between theory and practice this paper seeks to marry the academic debate over software patentability with concrete Canadian perspectives from inside the industry. It is the author's hope that this investigation will create a more stable practical foundation for legislative and policy based decision making in the future.[7] To this end the paper proceeds in five parts. In part one Canada's basic handling of software patents is examined - revealing a very undecided approach with far ranging practical consequences for numerous parties. This weak position, along with a dearth of Canadian based discourse in this area, leads the paper, in part two, to a comparative learning exercise involving the examination of U.S. and EU approaches to patent law. This establishes two paradoxically

[4] *Trade-Related Aspects of Intellectual Property Rights (TRIPS)*, WTO, 15 April 1994, Annex 1C of the Marrakesh Agreement Establishing the World Trade Organization at 27(1).

[5] James Bessen & Robert M. Hunt, "An Empirical Look at Software Patents" (2007) 16 Journal of Economics and Management Strategy 157 at 171.

[6] Due to its size, relative to the US, the sector has been dismissed as irrelevant in the academic, practical and legislative spheres. Interview of Legal Staff at Faskin Martineau (March 15, 2008).

[7] For example, the currently proposed revisions to MOPOP Chapter 13 – Computer Implemented Inventions. See http://www.cipo.ic.gc.ca/eic/site/cipointernet-internetopic.nsf/eng/wr00758.html.

divergent approaches operating under the international Trade Related Aspects of Intellectual Property (TRIPS) treaty.[8] Having established no clear policy direction via this comparison the paper turns, in part three, to an examination of the academic and theoretical discourse in this area. Strong arguments on either side of the patent innovation debate are examined but no clear answer emerges providing the paper with the impetus for a contextual investigation. Part four responds by examining the question of innovation from the perspective of three software companies with research and development offices in Canada. The trend which emerges is that Canadian software companies generally find software patents detrimental to their business objectives. Given the small sample size, confirmation of this trend within the broader Canadian software industry is not possible and further research is required in order to substantiate this papers recommendations. However, assuming that the perspectives of the software companies considered are representative of the Canadian software industry then the detrimental trend identified aligns the Canadian software industry with the non-innovation theories of software patentability explored in part three. By aligning this research with the legal and theoretical explorations undertaken previously the paper provides itself with the justification required for part five in which two potential policy responses are advocated, mainly: the exclusion of software from patentability or a carefully considered extension of patent law which responds systemically to the unique needs of the software industry. Most importantly, this paper establishes that before adopting a positive software patent regime the patent office, judiciary and legislature must present a clear and unified opinion based on the active contextual research of Canada's software industry and its relationship with patent law. The paper therefore establishes an approach, solution and justification for the correction of Canada's software patentability debate.

1.0 CANADA'S SOFTWARE PATENT PROBLEM

This section first introduces the core principles of patent law and how these core tenets affect software patentability. It then examines Canada's approach to software patents and the impact this has on the principle stakeholders in the software and legal industries.

[8] *Supra* note 4.

1.1 Patents Generally

In Canada the Patent Act provides the legal mechanisms through which patents are reviewed, granted and protected.[9] Patent law is viewed as a bargain between inventors and the state – in exchange for the inventor's full disclosure of the invention, the inventor is given an exclusive right to prevent others from making, selling, or using the invention for a fixed term. This bargain is viewed as a way to "…stimulate the creation and development of new technologies".[10] As such the notion of innovation tends to be the principle justification for patent regimes and is at the root of the software patentability debate.

Patent law was developed in Florence Italy and the pioneering Italian statute of 1478 introduced a set of core principles which form the basis of modern patent law today.[11] In Canada the Italian ideology is broken into four components: statutory subject matter, novelty, inventiveness and usefulness. To be patentable an idea and its implementation must meet all of the tests which flow from these tenets. The statutory subject requirement means that an invention must fall within the range of subjects which a state defines as patentable. Under Canadian patent we exclude from patentability all abstract theorems or mere scientific principles.[12] Novelty means that an invention must be original and not previously disclosed to the public. This requirement gives rise to the notion of prior art, that is disclosures to the public, to which patent examiners and inventors must be particularly wary.[13] Inventiveness is defined in Canada as that which would not have been obvious to a person skilled in the art.[14] Finally utility continues to be a vague concept in Canadian patent law as it is not defined by our statute. Courts have sometimes interpreted this to mean that a patent must have economic

[9] *Canadian Patent Act*, R.S.C. 1985, c P-4.
[10] David Vaver, *Essentials of Canadian Law: Intellectual Property Law: Copyright, Patents, Trademarks* (Concorde Ontario: Irwin Law Concorde Ontario, 1997) at 113.
[11] Ikechi Mgbeoji, *Global Biopiracy: Patents, Plants and Indigenous Knowledge*, (Vanvouver: UBC Press, 2006) at 16.
[12] *Supra* note 9 at s.27(8).
[13] *Supra* note 9 at s.28.2.
[14] *Supra* note 9 at s.28.3.

consequences, though others have taken a more liberal approach relying on the patent application itself to define its own utility.

1.2 Software Patents

In Canada software patents are not a specific form of patent, but rather the term simply identifies patents which use, or are related to, computer software. As such, software patents must still meet the traditional tests for statutory subject matter, novelty, inventiveness and utility. These core elements present some unique challenges in this field of innovation.

From a purely statutory perspective abstract theories, including mathematical ones, are excluded from patentability under Canadian law. However, there is no distinction, other than representation, between computer code, and mathematics. Software and computer code are simply human readable implementations of lambda calculus – a form of pure mathematics. The famous Church-Turing thesis established in 1936 that any computer function is simply the equivalent of a mathematical expression.[15] Thus, although computer programs are not explicitly excluded by Canadian patent law, there is a convincing argument that they fall dangerously close to the abstract theory exception. It helps to imagine, as developed by Ben Klemens, a spectrum of inventions. At one end we can place patentable physical machines made of transistors and diodes; at the other end we can place pure un-patentable math. Policy should select a clear dividing line between the patentable and the un-patentable at some point in this spectrum.[16] Problematically, separating machine from math rapidly becomes a grayscale exercise as they are often intimately related to one another.[17]

Novelty also presents an interesting problem. In most cases novelty is interpreted by Canadian patent examiners as requiring a thorough search of the U.S. Patent and Trademark Office's (USPTO)

[15] Douglas R. Hofstadter, *Godel, Escher, Back: An Eternal Golden Braid*, (New York: Basic Books, 1979) at 428.

[16] Ben Klemens, *Math You Can't Use* (Washington: The Brookings Institute, 2006) at 44.

[17] Grayscale is reference to both the 256 shades of gray traditionally recognized by computers, though almost indistinguishable by humans, as well as the shades of grey dividing machine from math in the sphere of software.

databases. The USPTO is recognized as the world's single largest prior-art resource and though prior-art searches should proceed further the cost and expediency requirements of patent examinations usually make this impossible.[18] Problematically software development and innovation has been occurring for much longer than we have had software patents. As a result prior-art in this field exists in a multitude of places.[19]

The vagueness of utility in Canadian patent law also creates difficulties as the product of a software program is not normally tangible. Unlike industrial patents where the monopoly is against a manufacturing process a software patent is held against a method or mathematical implementation of a concept. These ideas can be realized by anyone with access to a home computer. This raises the question of whether patents, which were intended to promote industrial innovation through economic incentives, have any role in constraining the actions of individuals.[20]

The requirement of inventiveness has a close link with that of novelty. The test for obviousness is a highly subjective one based around the "skilled person in the art".[21] As a relatively new field of technology computers are often bewildering to the average user – but to a skilled person in the art most software development is the obvious implementation of logical decision making processes. Worse still is the fact that most software development is simply the modification of existing prior art - an infinite regression of ideas built upon other ideas with no discernable starting point.[22] In computer science this is called code re-use and it is part of the developer's creed. Patenting software therefore means one patents a multitude of previous works with implications both up and down the innovation chain.

---

[18] *Supra* note 16 at 74.

[19] Grant C. Yang, "The Continuing Debate of Software Patents and the Open Source Movement" (2004) 13 Tex. Intell. Prop. L.J. 171 at 186.

[20] Russell McOrmond, "A Review of Software Patent Issues, Digital Copyright Canada" online:<http://www.flora.ca/patent2003/software-patent2003.shtml>.

[21] *Beloit v. Valmet Oy*, [1986] 8 C.P.R. (3d) 289.

[22] Stephen Hawking, *A Brief History of Time,* (New York: Bantam Book, 1988).

1.3 Canada's Basic Non-Position

The foregoing establishes that there are some conceptual problems with simply applying existing patent law definitions to computer software. Canada is not alone in attempting to recognize these theoretical difficulties. The result, however, is a disjunction between the legislative, practical and judicial treatment of patent laws as they apply to computer software. [23]

From 1978 to 2005 the Canadian patent office adopted the official position that patents for computer programs were not appropriate based on fears that they would hinder progress in an emerging field.[24] In 2005 the Manual of Patent Office Practice (MoPOP) was amended substantially, effectively reversing this position in a two phase re-interpretation of the Patent Act's statutory subject matter exception.[25] First, section 12.04.05 of MoPOP was added – holding that computer programs would be considered statutory subject matter so long as they were "… integrated with traditionally patentable subject matter". In addition to this categorization of software MoPOP introduced chapter 16 - an entirely new chapter on computer implemented inventions. In the context of software innovations only chapter 16 further clarifies the statutory definitions used in chapter 12 emphasizing that traditionally patentable subject matter may include not only physical implementations and results but any "…essentially economic result relating to trade, industry or commerce".[26] Then finally in a caveat it notes that this "economic result" requirement is not met simply by performing calculations producing useful information – thereby arguably excluding most traditional software programs.

Further changes to chapter 12 of MoPOP were introduced in 2009 and changes to chapter 16 of MoPOP are currently pending. As of this writing Chapters 12 and 16 of MoPOP do not accord with one another. However, the emerging subject matter requirement for computer programs appears to be that "the device must provide a

[23] The EU has been left in a vastly similar position as explored in part 2.3.

[24] *Supra* note 10 at 129.

[25] "Canadian Intellectual Property Office - Manual of Patent Office Practice", March 2007 at c.12 and c.16. MoPOP is a non-binding practical guide for practitioners and patent examiners which helps to explore the interpretation of the Canadian Patent Act.

[26] *Ibid.* at c. 16.03.02.

novel and inventive technological solution to a technological problem".[27]

Even for professionals the resulting legal terrain is far from clear. Lawyers with Bereskin and Parr, a leading Canadian intellectual property firm, opined recently that the MoPOP's clarifications mean little without additional jurisprudence.[28] Rather than sending a clear signal regarding patentability the patent office has established that software may be patentable if it meets certain criteria, is claimed appropriately, and produces an economic result. No further guidance is provided to anyone. Compounding this problem is the fact that the only case related to software patentability in Canada is from 1981 and it effectively held that computer programs are not patentable.[29] Consequently, the disjunction between legislative, practical and judicial treatment becomes clear. Statute and case law suggest that software is not patentable while the practical guidelines suggest that it may be patentable under carefully constructed circumstances. In such a climate it is difficult to trust that an issued patent would be of any concrete legal value. The result is what might best be termed a non-position. There is no unified agreement with respect to software patents between any of the governmental bodies responsible for upholding our patent laws. Far from clarifying the state of software patents in Canada the MoPOP guidelines simply created a great deal of breathing room for argument. What becomes evident is that legislative and judicial clarification of Canada's official position on these matters is required specifically in light of the fact that MoPOP is merely a set of interpretive guidelines.

1.4 The problem for parties

Given the complexity of understanding Canada's position on software patents it should come as no surprise that this has substantial practical impacts on numerous stakeholders in the software industry. At least five interrelated groups and their interests are impacted by

---

[27] Canadian Intellectual Property Office - Manual of Patent Office Practice", February 2010 at c.12.06.06b.

[28] Sam Frost and Ebad Rahman, "How to Protect Software Inventions", (2007) Managing IP 53 at 53 online:<www.managingip.com>.

[29] *Schlumberger Canada Ltd. v. Commissioner of Patents*, [1981] 56 C.P.R. (2d) 204 (FCA).

the ambiguous results examined above. First and foremost perhaps are lawyers who without further guidance are unsure of what is and is not patentable. As a direct result they are unable to advise their clients in any meaningful and practical manner.

Without clear direction from its legal representation the software industry in Canada doesn't know how it should proceed. Filing a patent is an expensive process and without clear guidelines firms are unlikely to pursue this avenue of IP protection. Worse still, due to Canada's previously long standing exemption of software from patentability companies may still labour under the belief that this is the law. The situation is further exacerbated by the fact that companies and their legal representatives may not discuss the implications of software patentability in other countries, such as the U.S. and EU, due to the influence of domestic confusion on the subject area as a whole.

This has a trickledown effect on software developers and lower level employees within the software industry. As this paper examines later companies do not appear to educate their staff regarding even basic intellectual property matters. Knowing that software is often the compilation and modification of existing code this results in a problematic environment within which developers are unaware of even basic legal issues regarding the licensing and patent protection of publicly available works.

Finally, given that MoPOP is the only source of true guidance in this area it is difficult to suggest that patent examiners have a clear understanding of what is and is not patentable in the software industry.

1.5 Summary

This section has established that there are issues with simply applying traditional patent law definitions to computer programs. In exploring the impact of these issues through Canada's existing statutes and guidelines it was shown that Canada has an uncertain and untested position regarding software patentability. The lack of agreement between legislative, judicial and practical approaches to software patenting has a substantial impact on numerous industry stakeholders and legal professionals. A substantial burden therefore appears to fall on the legislative and judicial bodies of Canada to

clarify the situation by responding to MoPOP's guidelines. In making these policy decisions it would appear worthwhile to investigate the responses and success of other countries with respect to software patentability. The paper therefore turns now to just such a comparative exercise.

2.0 LEARNING THROUGH COMPARISON

This section looks to the laws and experiences of other countries with the objective of evaluating and learning from their experiences. It begins with an examination of how TRIPS engages the issue of software patentability, thereby binding its signatories. It then examines the paradoxically divergent approaches of the U.S. and EU.

2.1 Requirements of TRIPS

The objective of the TRIPS agreement, signed in 1995, was to harmonize the patent systems of World Trade Organization (WTO) members in order to facilitate the protection, trade and secure exchange of intellectual property.[30] The agreement therefore binds Canada, the U.S. and EU nations as all are members of the WTO. With respect to computer programs article 27 (Patentable Subject Matter) of the agreement is most often cited as requiring that TRIPS signatories must extend their patent regimes to protect software. This section states that "patents shall be available for any inventions… in all fields of technology provided that they are… capable of industrial application".[31] TRIPS also states, in article 10 that "computer programs, whether in source or object code, shall be protected as literary works (copyright) under the Berne convention".[32]

These two sections create an interesting internal contradiction with respect to software. It has been argued that under TRIPS a given intellectual achievement should only attract one form of protection. Since TRIPS explicitly provides that computer programs are protected under copyright they shouldn't be protected by patent law under

[30] Daniel Gervais, *The Trips Agreement: Drafting History and Analysis* (London: Sweet and Maxwell, 1998) at 11.
[31] *Supra* note 4 at a.27.
[32] *Supra* note 4 at a.10.

article 27.[33] The converse of this argument is that TRIPS simply provides a minimum level of protection under article 10 – permitting individual nations to decide what subject matter should be patentable under article 27. On this view computer programs are more than simply lines of code but have functional aspects.[34] TRIPS in and of itself therefore does not appear to definitively answer the question of software patentability.

2.2 Development of U.S. Patent Law

The U.S. approach to software patentability has been one of slow historical growth principally through judicial decision making. The case of *Diamond v. Diehr* is normally identified as the case which tilted the U.S. towards its pro-software patent stance. In that case the Supreme Court held that "… an invention is not necessarily un-patentable simply because it utilizes software".[35] More recently *State Street Bank* is viewed as entrenching and firmly establishing the U.S.'s acceptance of both software and business method patentability.[36]

Like Canada the U.S. retains an exception for patentability in relation to pure mathematical algorithms but American courts interpret this exception very narrowly allowing software to be patented so long as it has any useful application. U.S. and Canadian statutory law are therefore almost indistinguishable. In addition, there is a striking similarity between MoPOP and the U.S. patent examiners guidelines, the Manual of Patent Examining Procedure (MPEP), on the subject of software. The substantial difference between Canada and the U.S. is that judicially the U.S. has seen far more treatment of patent cases at the Federal level. The country has thus been able to establish a clear and well publicized stance on the matter allowing practical application of the legislation.

The situation for the U.S. is, however, not devoid of problems. A rapid increase in software patent volume has resulted in an

[33] FFII Workgroup 2004, "The TRIPS Treaty and Software Patents" online:<http://eupat.ffii.org/analysis/trips/index.en.html>.

[34] Miguel E. Sciancalepore, "Protecting New Technologies in Latin America: The Case for Computer Software Patents in Argentia", (2006) 38 U. Miami Inter-Am. L. Rev. 349 at 375.

[35] *Diamond v. Diehr*, 450 U.S. 175 (1981) at 177.

[36] *State Street Bank v. Signature Financial Group,* 149 F.3d 1368 (1998)

overtaxing of the USPTO. The quality level of approved software patents is generally recognized as extremely low with a substantial number of patents being recognized as overbroad within the industry.[37] Because patents are presumed valid it is extraordinarily difficult and expensive to overturn them. As a result a variety of techniques have sprung up in the U.S. market to profit from this state of affairs - patent trolls being the preeminent example. Trolls game the system by first obtaining overly broad software patents then using those patents offensively to induce licensing fee agreements from companies theoretically infringing the patent. Companies submit to such extortion as the license fees requested are often less than the cost of fighting the patent. The USPTO has moved to rectify these problems as of late by attempting to hire computer science graduates so as to improve its prior-art searches and obviousness examination procedures.[38]

2.3 Development of EU Patent Law

In the European Union (EU) patent law is governed by the European Patent Convention (EPC) which was developed by the European Parliament (EP) – a legislative branch of the EU. Patent examination is performed by the European Patent Office (EPO). The objective of the EPC was to harmonize the securing of patent protection across national boundaries in Europe.[39] A patent granted by the EPO is therefore presumptively valid in every EU country. Nonetheless, individual countries are not bound by EPO decisions and are free to invalidate nationally challenged patents.[40]

Article 52(2) of the EPC specifically excludes from patentability not only scientific and mathematical theories but programs for computers. However, excluded subject matter under

[37] *Supra* note 16 at 73.

[38] "Intellectual Property: USPTO Has Made Progress in Hiring Examiners, but Challenges to Retention Remain" *General Accounting Office Reports & Testimony Newsletter* (1 August 2005).

[39] *Supra* note 19 at 180.

[40] John Moetteli, "The Patentability of Software in the U.S. and Europe", (Presentation for the Institut fur Europaishes and Internationales Wirtschaftsrecht St. Gallen Switzerland 28 October 2005) at 7 online< http://www.patentinfo.net/patentsearchersnet/download/THE_PATENTABILITY_OF_SOFTWARE_IN_THE_US_AND_EUROPE.pdf>.

52(2) is subject to article 52(3) which states that exclusions only operate where the "…patent relates to [excluded] subject-matter *as such*".[41] Interpretation of the words "as such" has caused deep divisions between the EPC, EPO and EP for over a decade.

Through its examination guidelines the EPO has voiced its opinion that the words "as such" are to be narrowly construed. The EPO thus holds that "…while computer programs are excluded by the EPC, software is not excluded subject matter if it…. brings about a technical effect". The EPC then goes on to define technical effect as anything "achieved by the internal functioning of a computer… under the influence of a… program".[42] The effect is that so long as one claims the "use" of a computer program rather than the programs "method" the EPO will grant a patent for the software.[43]

In 2005 the EPO moved to formalize this position through a bill before European Parliament. The EP responded by throwing out the bill and re-iterating that software programs were not to be viewed as patentable subject matter.[44] Since then the EPO has continued to approve software patents, but individual nations within the EU, namely Germany and the UK, have invalidated the patents challenged within their borders.[45] Practically speaking this means that software is patentable in the EU but that such patents will not be upheld before national courts in the largest EU markets. Echoes of the confusing Canadian situation are unmistakable, however, there appears to be a greater consensus between parliament and judiciary within the EU as compared to Canada suggesting that the EPO is simply acting as a renegade inside a system which generally opposes software patentability.

---

[41] *Convention on the Grant of European Patents (EPC)*, European Parliament, 5 October 1973 at 52.

[42] "The TRIPs Treaty and Software Patents", (Paper for FFII Workgroup, 2004) online:<http://www.eupat.ffii.org/analysis/trips/index.en.html>.

[43] Canadian patent attorneys will be familiar with this logic as an adaptation of the method versus use claim approach to medicines.

[44] R. Hilty & C. Geiger, "Patenting Software? A Judicial and Socio-Economic Analysis" (2005) 36 International Review of Intellectual Property and Competition Law 615 at 617.

[45] For an example of invalidation in Germany see http://ipgeek.blogspot.com/2007/11/german-federal-patent-court-invalidates.html.

2.4 Reconciling the Difference

The solutions of the U.S. and EU are generally divergent with the U.S. taking a pro-software patentability approach and the EU, at least at the legislative and judicial levels, calling for non-patentability. These variations in approach become almost contradictory in light of the fact that the U.S. and the EU are both signatories to TRIPS. The malleability of the TRIPS language effectively leaves nation states to their own devices in adopting or discarding software patentability. Therefore TRIPS, in and of itself, is of little assistance in helping us to clarify the Canadian state of affairs. What we can conclude from the U.S. and EU evidence is that the involvement of the judiciary and legislative branches is critical in establishing, publicizing and solidifying a strong position regarding software patentability – regardless of which position is assumed. This evidence, however, falls well short of helping us to establish a justifiable policy position. In response the paper now turns to an examination of the academic discourse in this area in order to determine if legal theory can provide us with any clearer answers.

3.0 INNOVATION OR SUBSTITUTION

In this section a brief overview of academic discourse in the area of software patentability is undertaken. Much of the literature reviewed originates in the U.S. as it is the principle world market which has adopted computer patents. This provides Canada with an excellent test bed from which we can learn about the impacts of such an economic experiment.[46]

3.1 Software Patents and Innovation

The correlation between innovation and patentability is at the center of the software patent dispute and a great number of academic papers deal with this subject. A summary of the literature in this area is perhaps best established by Robert Merges who states that despite initial concerns software "… patents have not killed software…. [but] this is hardly a ringing endorsement for the new regime."[47]

[46] *Supra* note 5 at 158. Bessen and Hunt describe the U.S. model as an experiment.
[47] Robert Merges, "Software and Patent Scope: A Report from the Middle Innings", (2006) 85 Tex. L. Re. 1628 at 1633.

Unpacking the history of this statement serves as a good exploration of the innovation question.

It was initially believed in the early 90's that the introduction of software patents would shut down innovation in the software industry or concentrate power in the hands of only a few companies.[48] Empirical evidence over the last 15 years suggests this simply hasn't happened – innovation and new market entrants continue to abound.[49] Innovation incentives are regularly cited by the U.S. judicial system as the motivating factor behind the extension of patents to software.[50] Though the innovation-patentability question is superficially simply, and attractive as a justification, drawing a conclusive link between the two has been a very elusive task. Bessen and Hunt, for example, manage to positively correlate R&D expenditures with the number of patents a firm owns.[51] Unfortunately, they note that there are numerous other factors besides the incentive to innovate that may create this statistical association.[52] Bessen and Hunt's research is also criticized for its reliance on outdated information as well as its use of data from irrelevant industry sectors such as manufacturing.[53] As a result even their limited findings have questionable application in a pure software market.

In a less cautious paper Grant Yang concludes that software patent incentives motivate both small and large companies to innovate.[54] However, his paper is based on second hand observations and pure theory rather than substantive research. His conclusions are also directly disputed by ongoing statistical research conducted by Ronald Mann et al. which finds that patents are of limited or even negative value to start-up companies.[55]

---

[48] "The League for Programming Freedom - Against Software Patents", Letter to USPTO, *USPTO* (24, October, 1990).
[49] *Supra* note 47 at 1634.
[50] *Supra* note 16 at 44.
[51] *Supra* note 5 at 173.
[52] *Ibid.* at 184.
[53] Robert Hahn and Scott Wallsten, "A Review of Bessen and Hunt's Analysis of Software Patents", (2003)
online:<http://www.researchineurope.org/policy/hahn_wallsten.pdf > at 2.
[54] *Supra* note 19 at 195.
[55] Ronald Mann & Thomas W. Sager, "Patents, Venture Capital and Software Start-Ups" (2006) 36 Research Policy 193.

Historically the link between innovation and patentability has been seriously questioned and the same appears no less true in the current debate.[56] As the basis for judicial treatment, and thus the US's approach to patent law, the innovation argument is therefore suspect. Given the diversity of economic factors at play one of the only ways to confirm this causal connection is to undertake a contextual review and directly survey the opinions of industry representatives.[57] Ronald Mann undertook just such a review in the U.S. and concluded that "… absent some other benefit all [software] firms would be better off saving the costs of obtaining patents". Once again this isn't a glowing review of the regime adopted by our neighbors. It does however lead to the question of whether patents serve some other function, besides as an innovation incentive, and whether the value of such a function makes software patentability worthwhile.

3.2 Software Patents as Something Else

The entrenchment of software patentability in the U.S., as well as the lack of conclusiveness regarding the innovation-patentability question, has driven Mann, Merges and others in recent research to accept rather than fight the regime. Instead of looking for justifications they now look to improve and explain the alternative value(s) created by its existence. Most notably they have examined the value of patents in the venture capital process. Venture capital is a particularly important element in bringing many software projects to market because innovation often occurs within small cash strapped companies who cannot afford the cycles of production and marketing which follow innovation.[58]

In the venture capital sphere Mann has convincing and repeatedly established that venture capitalists consider patents to be one of a number of factors used to establish the investment value of a

[56] "Debunking Software Patent Myths", (1992) Communications of the ACM online< http://www-swiss.ai.mit.edu/6805/articles/int-prop/heckel-debunking.html#NinePats>.

[57] P. S. Petraitis, A. E. Dunham & P. H. Niewiarowski, "Inferring Multiple Causality: The Limitations of Path Analysis" (1996) 10 Functional Ecology 421. Mathematics and statistics are naturally limited in this manner, though there are techniques for modeling multiple causality issues.

[58] *Supra* note 19 at 196.

firm.[59] In Canada anecdotal evidence suggests that the same is true, at least within the wider IP and venture capital fields, but that value differs substantially on a case-by-case basis.[60] Thus, although venture capital has largely shaped the U.S. software industry its relationship to patents, like the innovation question, is dependant on a wide range of factors.[61] This makes it difficult to establish any clear positive effects.

The consequences of the U.S. patentability model have nonetheless been felt by the software industry. Hunt notes that in the U.S. innovators who cannot afford to bring their products to market themselves will often resort to "… increasing their patenting in order to tax the rents earned on rival's inventions and to mitigate similar behavior in their rivals". These two approaches are more generally known as the offensive and defensive use of patent portfolios.[62] Offensive use refers to the sword like use of a patent by a company to challenge a competitor via claims of infringement. Similar to patent trolling offensive use requires firms to seek out targets and threaten legal action. Defensive use refers to the functioning of patents as a protective shield used to hopefully mitigate the effects of offensive claims. The entirety of this process, as noted in our discussion of patent trolls, is exacerbated by the issuance of overly broad patents. The end result is what has been termed the patent thicket. A jumble of overly broad and likely invalid patents held up by software companies to shield themselves from the market effects enabled and permitted by the regime.[63] Whether this result should be viewed as a benefit of the patent system is highly questionable.

First-to-market advantages are a further component of the larger economic impact created by software patents. The concept of first-to-market, also known as network effect, refers to the tendency in software for the first innovator with a publicly released product to become the market leader. Network effects in the software industry

[59] Ronald Mann, "Do Patents Facilitate Financing in the Software Industry?" (2004) 83 Tex. L. Rev. 961 at 966. AND J. Allison and A. Dunn and R. Mann, "Software Patents, Incumbents and Entry", 85 Tex. L. Rev. 1580 (2007)

[60] Marc Castel "IP Barrier or Enabler to Innovation" (Presentation to Challenges in IP Class, 24 March 2008) [unpublished].

[61] *Supra* note 47 at 1642.

[62] Robert Hunt, "When Do Patents Reduce R&D?" (2006) 96 American Economic Review 87 at 87.

[63] *Supra* note 16 at 83.

are intensified by at least two forces. First, product development lifecycles are extremely short.[64] Working products can be released within months. Second, and in opposition, the issuance of patents is regularly an 18 month procedure – plenty of time for a competing company to develop and release a product thereby gaining an important market lead.[65] As a result, diverting time and effort during R&D to patent ideas increases costs and reduces the ability to capitalize on the value of a first-to-market release. Consequently, the unique features of the software industry and market suggest that patents have a dubious value in this regard as well.

3.3 The Failure to Reconcile and Contextualize

The forgoing has established that software patents don't appear to have a clear connection with innovation. So far as research can say there is at best a positive correlation between R&D and patenting activities. The motivation behind such activities could be the result of numerous factors. If patents are not serving as an incentive to innovate then a justification for their existence needs to be based on some other set of benefits. Research into alternative benefits in the U.S. shows that they appear to have some value as alternative measures of a companies value and are thus sometimes beneficial in venture capital or Merger and Acquisition (M&A) situations. Patents also enable companies to engage competitors through alternative business models based on offensive and defensive tactics. Taken as a whole, even if these benefits were substantial, which they don't appear to be, they would need to be corroborated in the Canadian context. Keeping in mind these arguments the paper is thus lead, as was Mann, to consider contextual research as the solution to the inconclusive evidence presented thus far.

4.0 EXAMINING THE CANADIAN CONTEXT

Legal theory, statistical research and international approaches have provided us with no clear justification for a policy based decision regarding software patentability. The objective of this section is to move beyond these theories and debates using practical and

[64] *Supra* note 16 at 87.
[65] *Supra* note 10.

contextual Canadian evidence. The hope is that this will allow us to more clearly define the elusive innovation-patent relationship. In what follows profiles for three software development firms with innovation offices in Toronto are utilized as the basis for a contextualized profile of the Canadian software development industry and its relationship with patent law.[66]

4.1 Three Case Studies

Appendix A more closely examines the detailed responses of three Canadian software development firms to numerous questions about innovation and patentability. The objective of this section is simply to summarize the basis, limitations and conclusions which emerged from this research.

The company profiles available in Appendix A are based on personal interviews conducted and initiated by the author through industry contacts. Great efforts were made to minimize bias and obtain data which is representative of a range of sub-industries (software systems, software tools, web technologies) from companies in a variety of developmental stages (privately funded, seeking investment, publicly traded) and of various institutional sizes (from 40 to over 3000). Due to the limited sample size is difficult to say that the three companies form a representative sample. However, given the substantial agreement between them regarding the value of software patents it is argued that they serve as a valid starting point for understanding the Canadian context.

The responses of firms exhibited the natural tendency to break down along five principle lines of discussion within which the impact of patents can be examined, mainly: innovation, use, value, cost and education. With respect to innovation all three companies were asked whether they would continue to do business if software patents were unavailable. They all responded that their businesses would continue to operate substantially as they do today regardless of the status of software patentability.

On the topic of patent use only the U.S. based and largest company profiled admitted to the acquisition and defensive, but not

[66] For privacy, business and legal reasons the identities of these businesses are not revealed.

offensive, use of patents in order to establish bargaining chips for the purposes of cross licensing.[67] The other firms did not feel that software patents would serve any valid use in the software industry due to the impact of first-to-market factors.[68]

Patent value and cost were highly interrelated in the minds of the firms interviewed. The general consensus was that patents could provide valuable defensive protection in cross licensing agreements or infringement cases but that such value was generally outweighed by the cost of obtaining and maintaining a patent portfolio. Alternatives such as circumventing patent claims or halting product development were viewed as more viable responses to infringement claims.

Across the board the education of individuals regarding IP issues was found to be very low. Again only the largest and U.S. based company had any formal IP instruction program in place. At the opposite end of the short spectrum Company 1 was lacking even the vestiges of such a policy.[69]

In summary, the consensus among companies was best put by the IP director of one firm who stated that "… software patents are primarily a cost, a burden and a distraction to our business… [they] are an infrequent after-thought; our business focus is getting solid solutions to market first".

4.2 Alignment with Non-Innovation

The research suggests that all three companies would remain in business regardless of the status of software patents in Canada, or indeed the US. To this extent their responses appear to echo the work and opinions of Mann who "… doubt[s] that legal rules granting protection have a sufficiently substantial effect to alter the course of innovation in either direction".[70] The Canadian responses are also in line with Hunt's empirical research which concluded that "… growth in software patents may not be associated with an improvement in the incentives to innovate particularly in the [pure software] industries."[71] At best this describes a situation in which

[67] See Company Profile 3 in Appendix A.
[68] *Supra* note 19.
[69] See Company Profile 1 in Appendix A.
[70] *Supra* note 59 at 966.
[71] *Supra* note 5 at 184.

patents don't contribute to innovation in any substantial fashion. This raises the question of whether, through their establishment, additional positive or negative factors are introduced to the market by software patent regimes.

Our exploration of the U.S. model as well as patent laws core tenets established that software patents are plagued with problems in practical application. They have given rise to entirely new business models in the U.S. which at best have nothing to do with innovation and at worse impede innovation by exploiting the resources of companies that would otherwise invest further in R&D. The claim by Company 1, that it would consider ceasing operations in the face of any substantial lawsuit, supports the view that patents may negatively impact innovation under certain circumstances. In addition the USPTO has publicly recognized that its weak examination procedures have resulted in overbroad patents being issued. The result for companies caught in infringement claims is financially significant whether they settle or dispute the claims. Armed with this knowledge it becomes clear that if Canada does embrace software patentability it should be prepared for the additional burdens which will be placed on its patent examination officers and market players/innovators.

Given the low innovation value ascribed to software patents by the companies interviewed the conclusion that the Canadian context supports non-patentability is well justified. This is specifically true in light of the potential downside risks, such as trolling, exhibited in the U.S. and echoed as concerns domestically.

4.3 Patents as an Alternative Vehicle

Research in the U.S. did show that patents could hold value for companies seeking investment capital.[72] In such situations patents can be used as an alternative vehicle and measure of future market success. Though this does appear to occur in the U.S. the same was not found to be true in Canada.[73]

One company in the research panel was actively seeking investment capital and it did not see any substantial correlation

[72] *Supra* note 59 at 966.
[73] *Supra* note 55 at 207.

between investment potential and its patent portfolio.[74] This is again in keeping with Mann's research and the impact of first-to-market effects – the view was that more profit could be obtained by adhering to a rapid innovation and release schedule and such a model did not mesh with the lengthy patenting process.

The companies also dismissed or disliked the use of patents as defensive and offensive tools generally regarding it as a burden that caused them to stray from core business objectives. As such the principle alternative benefits potentially derived from a patent regime do not appear to have substantial application in Canada at this time. Thus, as a policy position, the non-patentability of software again appears to be a justified response.

4.4 The Education and Low Level Development Problem

A critical phase in the M&A and venture capital processes is the inspection of IP holdings in order to establish potentially patentable ideas as well as to identify latent violations.[75] The latter factor can be deeply impacted by the actions of low level developers on a project. We have explored how the innovation lifecycle in software is an extremely rapid and incremental process that builds upon previous works. Our research also established that IP education in the Canadian software industry is weak at best. These three factors combine in a unique fashion creating the potential for patents to have a profoundly negative impact on business processes.

It is not difficult to imagine a situation in which a low level developer begins development of a new cutting edge feature by searching for assistance on the internet. She'll probably find help in the form of publicly available though potentially licensed or patented software.[76] Using this as her starting point she may then invest substantial time in the modification or extension of this code. The end result may well be the incorporation of a licensed or even patented piece of code into the core elements of her work. Most of

[74] See Company Profile 2 in Appendix A.

[75] This came out in discussions with IP officers and legal staff for several of the companies profiled.

[76] Many software projects are protected under the GNU public license or another form of public licensing model. These models normally prevent the use of code for commercial purposes.

the companies interviewed didn't concern themselves with such issues at all.[77] Others felt that the education of project managers was sufficient.[78] Managers, however, admitted that they normally based their opinions on the time it takes a developer to complete a task. But as the above example shows time is but one element of the development process - problematic code can be incorporated into a project inadvertently with little or substantial effort.

The education problem therefore poses an additional downside risk for software companies in a patent based regime. This risk is only theoretical and no direct evidence of problems was exposed through the research. Nonetheless, it weights in favor of a non-patentability policy or in the alternative suggests that a clearly disseminated stance on software patentability is part of a fully functional and sound regime.

4.5 Conclusions in the Canadian Context

In the Canadian context the above review of theory and practice suggests that software patentability provides no significant advantages to companies as either defensive, offensive or investment tools. The core question of innovation, though always difficult to answer conclusively, appears to be impacted little by the existence of software patents. What has notably emerged through this practically grounded exercise are the downside risks of patentability to the software industry in Canada. Given that there are no clear advantages, and that the EU generally operates successfully under a no software patentability regime, the negative impact of patents appears to tip the entirety of the ambiguous debate substantially towards a finding of non-patentability. If the risk of implementing a software patentability regime results in the highly arguable positive results that have been explored but also brings with it, practically as well as theoretically, harmful elements then the suggestion that Canada adopt a non-patentability model with respect to software appears to be logically defensible. The paper now turns to a formalization of this recommendation.

[77] See Company Profiles 1 and 2 in Appendix A.
[78] See Company Profile 3 in Appendix A.

5.0 POLICY APPROACH AND CONCLUSIONS

This final section provides some policy and process recommendations as substantiated by the theoretical, legal and practical investigations undertaken above. The policy approach advocated is comprised of three main elements: clear assertion, better contextual research and non-patentability. Should software patentability be the only viable regime for reasons beyond the scope of this paper suggestions are made regarding the implementation details of such a system.

5.1 Canada Must Assert a Clear Position

The chief suggestion made by this paper is that Canada's policy regarding software patentability must be clarified. Though our patent examination office has developed guidelines for software patent practitioners and examiners they are untested and as such are of limited real-world value. The result has been a substantial degree of confusion in the software and legal industries regarding the requirements and state of software patents in Canada. Based on the U.S. and EU models it is clear that a unified vision and approach to software patent law is required from the legislative, judicial and examination branches of the government. The EU in particular continues to face challenges and debate in light of the rogue actions of its examination offices. Canada faces a similar situation unless there is co-operation and agreement between all responsible parties.

Beyond simply determining a clear stance on patentability the public assertion and dissemination of this position plays a critical role in the continued development of Canada's software industry. Software patentability impacts low level developers and individuals - due to the ability to create code using readily available tools. In contrast industrial patents are traditionally harder and more expensive to implement. As a direct result there is an increased risk of infringement and litigation exposure within the public at large. Education on these subjects is therefore just as important as the decisions themselves.

5.2 Additional Contextual Research

This paper has shown that the Canadian software industry appears to respond in unique ways to the patentability dispute. Unquestionably the research relied upon as the basis for these conclusions is limited – the three firms interviewed may not be a representative sample. Nonetheless, the substantial agreement between these corporations suggests that an underlying theme and approach exists within Canada. What this paper suggests is therefore valid on its face but requires further validation.

In 2002 the CIPO undertook a similar examination of its MoPOP guidelines and in an effort to obtain contextual data issued a call for papers.[79] An access to information request made in 2005 showed that the only responses to this call were from big industry and large corporate legal firms.[80] The bias inherent in this sample is clear. The conclusion is that in performing contextual research the legislature, judiciary and CIPO should not be satisfied with evidence from self selecting members of the industry. A properly conducted and active research initiative is therefore advocated – a replication of this papers approach would be considered an appropriate starting point.

A proactive research endeavor is further supported by the inconclusiveness of international laws, national responses and legal theory in the software-patentability debate. This paper has shown that resolving these problems may only be logically justifiable if we look at the contextual response and impact to Canadian businesses in order to resolve the overall uncertainties which have been explored.

5.3 Non-Patentability Approach

The argument for a non-patentable software regime in Canada has been the focus of this paper. The reasoned and pragmatic approach to this claim has proceeded in three identifiable phases. First the legal responses of countries and theoretical discussions of patent law were shown to be inconclusive. Second, a research based investigation of the Canadian context was used to show that our industry generally sees little value in software patents. Third,

[79] See http://strategis.ic.gc.ca/sc_mrksv/cipo/patents/pt_notice-e.html.

[80] In accordance with the Canadian Access to Information Act such requests can be made by individuals for government documents. Details of the request and response can be seen at http://www.flora.ca/A-2004-00246/.

practical and theoretical negative factors were shown to be introduced into the market by software patent regimes. It is argued that these negative aspects, in light of the virtually balanced and persuasive arguments which otherwise exist, are enough to support a logical decision favoring the non-patentability of software in Canada. In short, and in Canada, the negatives which are potentially introduced outweigh the negligible positive impacts which were explored. Under such an analysis there appears to be no logical justification for the implementing a software patentability regime in Canada at this time.

5.4 Patentability Considerations

If adopting a non-patentability regime should provoke substantial backlash in areas beyond the scope of this paper, such as in international trade, the results of this research and analysis still serve to establish some important points. First, the goal of asserting a clear and unified position on software patentability remains critical to the establishment of practically useful laws and guidelines. There is no substantial difference between U.S. and Canadian laws what differentiates them is the level of judicial involvement. Nationally a high level of involvement has enabled the U.S. to establish a clear and well publicized stance on the matter allowing predictable and practical application of its legislation. This level of involvement is required in Canada. Second the undertaking of further contextual research retains value as it may provide further insight into the unique factors which shape the Canadian software industry and which should therefore guide any policy decisions.

Beyond these re-affirmations we have seen that the U.S. model is not a cure-all which should be adopted whole heartedly without closer examination. The Federal Trade Commission (FTC) in 2003 released recommendations of a similar nature stating that consideration of the harm to competition as well as the costs and benefits of implementation had to be considered before simply extending the scope of patentable subject matter. The FCC also advocated for a contextual and economic learning exercise similar to that which is proposed above and which was undertaken by this paper.[81] The FCC's recommendations, coming from within the U.S. which is assumed by most to have an unshakable pro-patent regime,

[81] *Supra* note 16 at 151.

serve to highlight the fact that software patentability is an ongoing international debate. The papers examination of the EU model has also shown that a more limited acceptance of the software patent may be acceptable – so long as we make the position being adopted clear. What is principally objectionable is allowing the current state of uncertainly to prevail due to its wide ranging practical impacts.

The extension of patentability to software may also have substantial impacts on CIPO. This position was supported by our exploration of the USPTO's problems with software patent examination and over breadth. The resulting emergence of patent trolls and new market models in response to these problems are issues which require attention. Klemens suggests that such issues may be overcome by additional patent requirements within the software application process including the filing of source code which should then be searchable by developers – thereby substantially reducing uncertainty and search costs.[82] In addition the problems created by patent thickets suggest that a new dispute process for patents may well be justified. In fact, searchable data banks and new dispute mechanisms might be useful in other patentability fields such as chemistry or biotechnology. As such software patents need not be a separate type of patent, but the requirements of usefulness, inventiveness and novelty may require that we adopt new documentation and submission policies across the board, or holistically, in order to accommodate new subject matter.

5.5 Final Concluding Remarks

Obtaining a consensus between the legislature, judiciary and CIPO on the subject of software patentability is no small objective. What this paper has hopefully shown is that Canada's continued vague and untested approach to software patentability has significant negative real world impacts. As a growing and important part of Canada's economy the software sector requires stable practical guidance. Through the approach of contextualized research the paper has overcome the inconclusiveness of theoretical and empirical research. In addition it has attempted to reconcile and learn from international approaches. Through this research lens an industrial environment which supports the non-patentability of software in

[82] *Supra* note 16 at 151.

Canada has emerged. The final result is both an approach and justified solution to the policy issues currently facing Canada with respect to software patentability. The limits of this research have been made clear, but the effort of this paper was to establish a starting point for the discussion of this matter in Canada – something which is notably absent today. In the process it has hopefully also added substantially to the existing debate and can act as a guide for future policy decisions in this area of innovation.

APPENDIX A

Company Profile 1

Basic Profile

The company is involved in the development of banking and wealth management software. The resulting products are large scale institutional software programs for investment firms. The company is privately funded as the side project for a much larger capital investment firm. Among the firms interviewed its size is moderate with approximately 80 employees the majority of whom are software developers. The companies principle clients are in the US.

On Innovation

The company sees itself as a transcriber of well established banking transaction rules into software. Because of the nature of the generally accepted nature of these rules it considers itself generally immune from patent infringement. As a result patents play no negative or positive role in the companies day-to-day affairs. The company unhesitatingly responded that with or without software patents in Canada it would continue to develop software.

On Patent Use

The company holds no patents and has never been sued or approached regarding patent infringement during its 20 year history. It therefore saw no use either offensively or defensively for patents in the software field. When informed that the leading case on business method patents in the U.S. was related to banking transaction software the company, though slightly unnerved, dismissed the issue under the belief that software was not patentable in Canada.[83] Lawyers for the company noted this as a risk in the software industry generally and one which has been discussed with the company but not pursued.

[83] *Supra* note 36.

On Patent Value

The company perceived no value in software patentability. Lawyers for the firm did perceive a negative value for the company should it ever face an infringement claim. With no patent portfolio it would be unable to cross license and would have to settle or dispute the infringement claim. Representatives conjectured that the company would likely cease operations if faced with any substantial patent lawsuit.

On Patent Cost

Software patents, as of yet, have not cost the company any substantial legal or licensing fees – though legal staff recognized this as a substantial area of liability given U.S. trends and the companies reliance on U.S. clients.

On Education

Lack of education regarding software patents became clear at two principle levels. First discussion between legal staff and management for the company was minimal. Furthermore, no discussion between management and development staff occurred regarding intellectual property rights and their products. In order to obtain any useful information regarding patents the author was directed to speak with the companies outside legal council – no one within the firm felt they could answer the questions posed.

Summary

Having been in this sector of software development for over 20 years, the companies general approach is one which is frozen in that history. Events in the U.S. concerning the core of their business have not been considered seriously by legal staff or the companies management. At best this places the company at risk of being pursued for patent infringement. Though patents do not reduce innovation at this company on a daily basis licensing or lawsuits could cause the company to cease its practices entirely. With no intention of changing its practices the company makes a clear statement in favor of software non-patentability.

Company Profile 2

Basic Profile

The firm develops applications for the interactive web 2.0 sphere. Their principle goal is to create engaging applications that are easy to use. They distribute their products via telecommunication providers. The company is just over five years old and though it has proven itself stable has been looking for investment capital throughout its lifetime. Employing around 40 individuals it is the smallest and most rapidly moving company profiled. It holds no patents and has never been approached regarding licensing or infringement. The company performs work globally with clients in the EU and US.

On Innovation

The company considers itself a leader in innovative broadband technologies and device integration. Though it understood the argument that patents could support innovation it does not operate in that fashion. Innovation was seen as stemming from the cutting edge and rapid work of its development staff. This was supported through income earned by being the first to market with products. Over the years the company has continually adapted to market changes in order to keep that edge and to maintain profitability. Without patents the company stated that it would continue to innovate and develop software in exactly the same way it does today.

On Patent Use

As innovators the company believed it could respond to patent infringement claims in one of two ways; either by circumventing the patent through the use of a different product or programming approach, or simply by dropping that line of business. As such patents were of no use either offensively or defensively to the company as they had no intention of entering into cross licensing agreements.

On Patent Value

The company perceived the value of patents to protect innovation but felt they were inappropriate for their business model for two reasons. First, the company felt it didn't have the time or resources to devote to patenting its ideas. Second, the patent process represents a time

hurdle both in terms of preparation and issuance. Being first to market for the firm was the critical factor in survival and this wasn't viewed as meshing well with the patent process.

On Patent Cost

Patent costs to the firm have been minimal. Given their perspective on the ability to avoid or shed parts of their business that were infringing patents they also don't perceive substantial costs arising from licensing or litigation.

On Education

Management had considered patent and intellectual property issues as they related to the company, but based on the feeling that patents were a non-issue had not disseminated information any further. As a result development staff were not aware of software patent issues – a concern in cutting edge development where a great deal of code is open source.

Summary

More than likely this company could continue to operate as it imagines due to its diverse portfolio of ideas and products. This therefore directly opposes the theory that patents stimulate innovation. If anything patents would only cause this company to drop lines of R&D – not adopt new ones. For a fast moving internet company patents were not perceived as providing any benefits.

Company Profile 3

Basic Profile

The company has over 3000 employees and is publicly traded and commercially successful. It is principally involved in the development of software tools for developers. As such its investments in R&D are substantial and deeply technical. The company is based out of California but after acquiring a smaller start up in Toronto five years ago it shifted much of it early stage R&D to Canada. The company holds numerous software and business method patents

related to its products. It has also been involved in several cross licensing and settlement related disputes.

On Innovation

Patents are viewed by the company not as something which promotes innovation but something which allows the company to fortify its position in the market. The companies nature is to identify business problems and build solutions to those problems. The overriding objective and focus is to deliver those solutions to market swiftly - patents are an infrequent after-thought.

On Patent Use

Representatives viewed the use of patents as primarily a defensive exercise in order to establish trading cards for cross licensing and infringement disputes with competitors and patent trolls.

On Patent Value

The company recognized that patents do enabled them to protect their R&D investments. However, they did not view patents as a revenue stream, but rather as an additional layer of armor against lazier competitors who prefer to imitate rather than to create. They also acknowledged that patents have a certain marketing value for customers who appreciate the notion of patented or patent-pending technology. Patents were viewed more generally as a disruption to the core business focus of software development.

On Patent Cost

The company had spent a great deal of time streamlining its patent process in order to reduce application and settlement costs. They did not believe that these costs had been fully offset by the value to the company of patents. Defending against patent infringement accusations has cost the company substantially both financially and through lost development time. The costs of patents were therefore viewed as necessary in the market place – an added cost of doing business.

On Education

The company has a well determined policy of informing project and team managers about patent and intellectual property issues. Team managers are then responsible for the auditing and validation of R&D efforts by their team. Team leads said they normally didn't bother development staff with IP concerns, but viewed themselves as involved enough to spot such issues.

Summary

With a department which handles IP issues this company clearly had the most mature and advanced approach to software patentability. The company attributed this both to its size as well as it U.S. ownership and headquarters. Nonetheless, even though it sees itself as operating within a software patentability regime it generally disagrees with the burden and cost imposed by such a system. As a software company it still sees itself as a small company surviving on rapid innovation and first-to-market successes. Patents, even when necessary to defend against external forces, were not viewed as fully functional under such a business model.