2006RP-04

The Legal Issues Surrounding Free and Open Source Software: Challenges and Solutions for the Government of Québec

Pierre-Paul Lemyre, Richard Willemant

Rapport de projet
Project report

Part of a RESOLL study conducted and translated for the Treasury Board of Canada Secretariat

Montreal
February 2006

CIRANO
Centre interuniversitaire de recherche en analyse des organisations

CIRANO

Le CIRANO est un organisme sans but lucratif constitué en vertu de la Loi des compagnies du Québec. Le financement de son infrastructure et de ses activités de recherche provient des cotisations de ses organisations-membres, d'une subvention d'infrastructure du Ministère du Développement économique et régional et de la Recherche, de même que des subventions et mandats obtenus par ses équipes de recherche.

CIRANO is a private non-profit organization incorporated under the Québec Companies Act. Its infrastructure and research activities are funded through fees paid by member organizations, an infrastructure grant from the Ministère du Développement économique et régional et de la Recherche, and grants and research mandates obtained by its research teams.

Les organisations-partenaires / The Partner Organizations

PARTENAIRES
. Alcan inc.
. Banque du Canada
. Banque Laurentienne du Canada
. Banque Nationale du Canada
. Banque Royale du Canada
. Bell Canada
. BMO Groupe financier
. Bombardier
. Bourse de Montréal
. Caisse de dépôt et placement du Québec
. Fédération des caisses Desjardins du Québec
. Gaz Métro
. Hydro-Québec
. Industrie Canada
. Ministère des Finances du Québec
. Pratt & Whitney Canada
. Raymond Chabot Grant Thornton
. Ville de Montréal

. École Polytechnique de Montréal
. HEC Montréal
. Université Concordia
. Université de Montréal
. Université du Québec
. Université du Québec à Montréal
. Université Laval
. Université McGill
. Université de Sherbrooke

ASSOCIÉ À :
. Institut de Finance Mathématique de Montréal (IFM^2)
. Laboratoires universitaires Bell Canada
. Réseau de calcul et de modélisation mathématique [RCM^2]
. Réseau de centres d'excellence MITACS (Les mathématiques des technologies de l'information et des systèmes complexes))

ISSN 1499-8610 (Version imprimée) / ISSN 1499-8629 (Version en ligne)

The Legal Issues Surrounding Free and Open Source Software: Challenges and Solutions for the Government of Québec

Pierre-Paul Lemyre, Richard Willemant

Abstract

The Government of Québec is slowly but surely turning its attention to the issue of free and open source software in response to the interest shown by Québec's software industry and the attention paid to the phenomenon by governments around the world. This openness is easy to understand given an environment in which online service provision to citizens must be enhanced while minimizing expenditures on technology, curtailing service providers' control over the administration, and promoting the development of the information society in Québec. Nonetheless, as we see in the news, adoption of this new attitude toward to software development is not always immune to legal challenges. Consequently, the manner in which Québec law interacts with free and open source software, as well as the associated risks, assume a particular significance.

The analysis we present here reveals that the law, as it currently stands in Québec, appears adequate to effectively address the various legal issues inherent in the use of free and open source software. First of all, no legal rule seems to be incompatible with the validity of free and open source licences, despite that fact that few of them were designed with the Québec legal system in mind. Moreover, both federal copyright rules and Québec regulations affecting contractual liability allow the authors and users of free and open source software to effectively preserve the freedom of computer code, which is typically the purpose of free and open source licences.

Nonetheless, it remains the case that some legal risks are associated with free and open source software. These risks may arise from the formalism requirements included in the Copyright Act, prior violations of intellectual property rights by third parties, or simply from the broader contractual protection afforded to licensors. Consequently, integrating free and open source software into the technology strategy of the Government of Québec requires setting up some initiatives to allow these risks to be mitigated as much as possible and to enable the management of those risks that cannot be completely eliminated.

1. Executive summary

[1] The Government of Québec is slowly but surely turning its attention to the issue of free and open source software in response to the interest shown by Québec's software industry and the attention paid to the phenomenon by governments around the world. This openness is easy to understand given an environment in which online service provision to citizens must be enhanced while minimizing expenditures on technology, curtailing service providers' control over the administration, and promoting the development of the information society in Québec. Nonetheless, as we see in the news, adoption of this new attitude toward to software development is not always immune to legal challenges. Consequently, the manner in which Québec law interacts with free and open source software, as well as the associated risks, assume a particular significance.

[2] The analysis we present here reveals that the law, as it currently stands in Québec, appears adequate to effectively address the various legal issues inherent in the use of free and open source software. First of all, no legal rule seems to be incompatible with the validity of free and open source licences, despite that fact that few of them were designed with the Québec legal system in mind. Moreover, both federal copyright rules and Québec regulations affecting contractual liability allow the authors and users of free and open source software to effectively preserve the freedom of computer code, which is typically the purpose of free and open source licences.

[3] Nonetheless, it remains the case that some legal risks are associated with free and open source software. These risks may arise from the formalism requirements included in the Copyright Act, prior violations of intellectual property rights by third parties, or simply from the broader contractual protection afforded to licensors. Consequently, integrating free and open source software into the technology strategy of the Government of Québec requires setting up some initiatives to allow these risks to be mitigated as much as possible and to enable the management of those risks that cannot be completely eliminated.

2. About the authors

[4] **Pierre-Paul Lemyre** is International Projects Manager at LexUM,[1] the legal software team of the Centre de recherche en droit public (the public law research centre) at the Université de Montréal. In this position he has developed and overseen many projects about the dissemination and management of legal information. His recent accomplishments include Droit francophone, the legal portal of the Agence intergouvernementale de la francophonie, JuriBurkina, the Legal information Center of Burkina Faso, and the report *Feasibility Study for a Telecommunication Decision Database*, completed for the World Bank and the International Telecommunication Union.

[5] A lawyer by training, he has been interested in free and open source software and its impact on the legal world for many years. His Master's thesis in information technology law was titled *Les logiciels libres sous l'angle de la responsabilité civile* (Free software from the perspective of public liability). He also had the opportunity, in 2000 and 2002, to be invited as a speaker at the conference *Rencontres mondiales du libre,* in Bordeaux.

[6] **Richard Willemant** is a member of the Paris bar association. A lawyer by training, he is currently working on an LL.M. (Master's of Law) degree at Montréal's McGill University in the Institute of Comparative Law.[2] In France, he holds a Master's (DESS) in Multimedia and Computer Law (Université Paris II Panthéon-Assas) and a Master's (DEA) in contract and property law (Université Paris I Panthéon-Sorbonne).

[7] His interest in free and open source software dates from a collaboration with Maître Christiane Féral-Schuhl, an associate in the Salans law firm (Paris), for whom he conducted an analysis of the validity and impacts of free and open source licences under French law. He also contributes to the l'ADIJ (Association pour le Développement de l'Informatique Juridique), which, among other things, deals with e-administration.

[1]LEXUM, source : <http://www.lexum.umontreal.ca>.

[2]Institute of Comparative Law, source: <http://www.law.mcgill.ca/institutes/icl/>.

3. Table of Contents

4. Introduction

[1] Like all societies, Québec depends to a large extent on the pooling of free and open resources. The most important of these resources is undoubtedly the French language. No one has an exclusive claim on the words, no one controls their use, and yet our language continues to evolve and develop to the benefit of many enterprises. The same can be said of scientific research and, more broadly, of public goods such as highways and municipal parks.[3]

[2] However, when confronted with the prospect that this same principle could apply to software, the vast majority of people are initially mystified, even stunned. It is easy for us to accept that language can be used freely and that the results of scientific research are made available to all, but many feel that software must necessarily be proprietary.

[3] However, this vision is not shared universally. The extent of the free and open source software movement in recent years illustrates that fact very well. In fifteen years, this new approach to software development has gradually transformed the software industry by making available reliable and robust applications, such as the GNU/Linux operating system, while simultaneously attracting billions of dollars in investment.

[4] Far from remaining on the margins, Quebeckers have contributed actively to the development of free and open source software. In all likelihood, this is one of the elements underlying the interest shown in the domain by the Government of Québec in the framework of its online government program. This openness to the free is easy to understand, given a context in which online service provision to citizens must be enhanced while minimizing expenditures on technology, curtailing service providers' control over the administration, and promoting the development of the information society in Québec.

[5] The Government of Québec is not the first to show an interest in, and to examine, the role it should play with respect to free and open source software. Thus, certain governments, such as those of Venezuela[4] and Canberra Territory in

[3]Lawrence LESSIG, *Free Culture*, Penguin Press, New York, 2004; Lawrence LESSIG, *The Future of Ideas: The Fate of the Commons in a Connected World*, Vintage, New York, 2002.

[4]Gregory WILPERT, "Venezuela's Public Administration to Use Open Source Software", (2004) *VenezuelaAnalysis*, source: <http://www.venezuelanalysis.com/news.php?newsno=1457>.

Australia,[5] have already adopted statutes requiring the public administration to use free and open source software whenever feasible. A bill with similar provisions has also been proposed in Peru.[6] Other countries, such as France, have created agencies with the specific purpose of integrating free and open source software into their administrations. Elsewhere, official policies recommending the use of free and open source software have been elaborated (European Union, Israel).

[6] However, these various governmental experimentations from around the world have demonstrated that this new attitude vis-à-vis software development can create difficulties. This is particularly true in the legal domain, for which recent events have provided numerous examples.

[7] Without question, the most celebrated of these examples is the legal dispute between SCO and IBM, which gradually spread its tentacles until it engulfed the entire free and open source software community. SCO, owner of the operating system UNIX, alleged that IBM committed several offences, including the divulgation of trade secrets and breach of contract, by integrating its proprietary code into the free and open source operating system GNU/Linux. Going even further, SCO took steps to collect royalties from 1500 companies using GNU/Linux, invoking a violation of its intellectual property rights. The most recent turn of events in this legal imbroglio has raised questions regarding the constitutionality of the General Public License[7] (GPL) in the United States—a licence under which SCO itself distributed its own operating system for several years.

[8] Questions about the validity of the GPL have also arisen in France, especially in academic circles. It has been deemed incompatible with French law owing to the fact that it is unilingually English, that it contains certain clauses providing for blanket release from liability, and the totally unrestricted scope of the rights granted. These observations have spurred three government agencies to announce the elaboration and dissemination of the CeCILL licence,[8] a free and open source licence adapted to

---

[5] *Government Procurement (Principles) Guideline Amendment Act 2003*, A.C.T. (2003), 63, s. 6A, source: <http://www.austlii.edu.au/au/legis/act/num_act/qpgaa200363o2003595/index.html#s1>.

[6] *Bill 3030,* tabled on May 29, 2002, by senator Luis Gonzales Reinos, source: <http://www2.congreso.gob.pe/Sicr/TraDocEstProc/CLProLey2001.nsf>.

[7] FREE SOFTWARE FOUNDATION, "GNU General Public License", 2005 *FSF*, source: <http://www.fsf.org/licensing/licenses/gpl.html>.

[8] CEA (Commissariat de l'Energie Atomique, the French atomic energy commission), CNRS (Centre National de la Recherche Scientifique, the French national centre for scientific research), and INRIA (Institut National de Recherche en Informatique et en Automatique, the National institute for research in computer science and control), "French Free Software licenses," 2004 *CeCILL,* source: <http://www.cecill.info/index.en.html>.

the French legal context. Though contested by the Free Software Foundation's (FSF) French branch, the odds are that CeCILL will soon play an important role in France, especially in the public administration.

[9] Furthermore, free and open source software is also attracting attention in the context of the elaboration of regulations governing software patents—an issue on which seas of ink have been spilled in Europe over the past three years. This is understandable, since the patentability of software is potentially very detrimental to free and open source projects, which do not always have access to the financial means required to fully benefit from patent protection, and the philosophy of which is fundamentally at odds with the appropriation of ideas. The efforts of free and open source software defenders have been fruitful to date, thanks to the support of Poland, since the most recently proposed directive[9] does not appear likely to be adopted by the Ministerial Council of the European Union.

[10] These various issues all lead to the affirmation that the definition of a government policy in support of the use of free and open source software must include a legal component. Consequently, the challenges and solutions described in this study are primarily designed to shed light on the legal context underlying the use of free and open source software by Québec's public administration. This analysis is all the more important since the law is intimately linked to cultural and social values, and thus how the law deals with free and open source software is likely to vary from one country to the next. In Québec, bijuralism adds a further specificity that must be taken into account.

[11] Being first and foremost software, free and open source software primarily benefits from the protection afforded by copyright, like proprietary software. Not being bound by any formality, copyright automatically exists following the creation of software. Furthermore, it is subject to uncountable divisions, a feature that is extensively used by the authors of software in light of the diversity of existing software licence models. The primary purpose of these licences is to cede to a third party some portion of the rights belonging to the owner under copyright law. Thus, the main distinction that can be observed between free and open source licences and other software licence types is in the scope of the rights granted to the users. While proprietary software licences restrict the use, modification, and redistribution of

[9]Proposal for a Directive of the European Parliament and of the Council on the patentability of computer-implemented inventions, COD 2002/0047, source: <http://wwwdb.europarl.eu.int/oeil/oeil_ViewDNL.ProcViewByNum?Lang=1&ProcNum=COD/2002/0047>

software as much as possible, and it is distributed in a binary form, users of free and open source software receive all of these rights along with the source code.

[12] Clearly, this innovative fashion of granting rights could not spread without being challenged. As a consequence, several authors have advanced arguments against the validity of free and open source licences in recent years. In general, these arguments revolve around: a contention that the objective of these licences is undermining the principles of copyright, the viral character of free "copylefted" licences, and the technique used to obtain consent from the licensees. Given these circumstances, it is vital to assess their validity under Québec law. Since copyright is a matter of federal jurisdiction in Canada, it is first and foremost the Copyright Act[10] with which free and open source licences must comply. However, software licences also possess the characteristics of a contract, so the relationships established between licensor and licensee are also governed by the general law of obligations under Québec civil law.

[13] Aside from the issue of their validity, free and open source licences raise some questions regarding the enforcement of the legal rules governing intellectual property. Consequently, it proves essential to establish the potential impact of non-compliance by the licensee with the terms of the licence if we are to understand its enforceability within Québec. Conversely, the SCO litigation motivates us to ask questions regarding the way in which rights are transferred between developers and the consequences for the licensee of a prior violation of the licence, or simply of its invalidity. Finally, the particularities of free and open source software give rise to two further legal difficulties, one pertaining to issues of compatibility between the various types of licence, and the other to their interaction with the software patent regime.

[14] The contractual liability of the parties raises further concerns in the matter of free and open source software. In Québec, assessing the obligations of the parties first implies associating the licence contract with one of the contract regimes specified in the Québec Civil Code (QCC).[11] Imputing liability to the licensee should generally be a simple matter, since free and open source licences are abrogated as soon as one of their clauses is violated. Under those circumstances, the licensee is automatically subject to copyright law penalties. Conversely, anyone wishing to

[10] *Copyright Act,* R.S. (1985) c. C-42, source: <http://www.canlii.org/ca/sta/c-42/>.

[11] *Québec Civil Code*, L.Q. 1991, c. 64, s. 1708 and f., source: <http://www.canlii.org/qc/laws/sta/ccq/index.html>.

impute liability to the licensor will need to take into account the implicit guarantees specified by the law, the clauses in the licence that limit liability, and the consumer law in effect.

[15] Given the state of the law in Québec, it appears possible for the Government of Québec to orient its technological strategy around free and open source software. Nonetheless, establishing a strategy allowing the optimal use of free and open source software, while reducing exposure to legal risks to a strict minimum, implies that certain initiatives must be taken. Among other things, these initiatives will allow the government to honour its contractual commitments to current service providers, to manage the legal context of the use, and redistribution of, free and open source software by the administration, to ensure consistency in the position of the government of Québec toward intellectual property, to share the inherent legal risks with other stakeholders and, finally, to comply with Canada's international commitments in the area of trade.

5. Free and open source software and the law

[16] In the past fifty years, the law has had to accommodate a new discipline, computer data processing, as well as the new types of works that characterize it. Furthermore, far from being a curiosity for legal scholars, software is governed by a set of relatively well-defined regulations today. It is primarily intellectual property rights law, and more specifically the *Copyright Act*, that determines the scope and terms of application of the rights of software authors. Since they have the authority to establish conventions involving these rights, as in the case of licences, contract law also plays an important role in this area. However, the nature of software licences varies enormously: Proprietary licences tend to limit the rights granted, while free and open source licences seek to expand them. Moreover, not all free and open source licences are created equal.

5.1. The legal protection of software

[17] Throughout the Western World, certain products of the human intellect are considered eligible for appropriation. The property right covering intangibles developed over the course of centuries to promote artistic creativity and the advancement of knowledge by guaranteeing a monopoly to inventors over the disposition of their work. From these rules follow the assumption that the author of software is the first owner of its intellectual property rights. Since software, once it has been captured on a medium, assumes a predominantly literary form, copyright rules apply to it.

[18] Internationally, copyright protection is governed by the *Berne Convention for the Protection of Literary and Artistic Works*[12] (the Berne Convention). This document establishes a minimal framework of protection for literary and artistic works to safeguard the interests of the author. This protection extends to the creative expression of the work, but not to its underlying concept. Thus, for example, the author of a program that performs a mathematical calculation possesses some rights to the chosen computational technique, but not to the algorithm on which it is based. In particular, he obtains the right of control over reproduction, transformation, and distribution of his work. While enforcement of copyright law remains a national matter, the Berne Convention extends copyright protection to the territory of its 139 member countries.

[12] *Berne Convention for the Protection of Literary and Artistic Works*, (2005) Juris International, source: http://www.jurisint.org/pub/01/en/147.htm.

[19] Canada is one of the countries having ratified the Berne Convention. Since copyright is a matter of federal jurisdiction,[13] implementation of this treaty in Québec law falls under the *Copyright Act* of Canada. This Act leaves no doubt as to its applicability to computer software, since its Section 2 specifies that software is assimilated to literary works. Furthermore, the broad definition of software in the Act should ensure that no computer program falls outside of its purview:

> "a set of instructions or statements, expressed, fixed, embodied or stored in any manner, that is to be used directly or indirectly in a computer in order to bring about a specific result."[14]

[20] However, for the author of software to be able to benefit from the protection afforded by the *Copyright Act*, it is essential that it constitutes an original work. With respect to this, the Supreme Court recently formulated new criteria regarding the exercise of a skill or judgement in a productive process in its ruling in the case of CCH Canadian Ltd. v. Law Society of Upper Canada.[15] Usually, in fact, it suffices that the software was created independently. Consequently, two programs that perform the same mathematical calculation may be eligible for this protection, provided that neither was copied from the other. The purpose, the source of inspiration, and even the computer code may be identical, the originality of each program depends primarily on the independent effort that went into creating it.

[21] Furthermore, to be eligible for protection, software must necessarily be captured on some type of medium, whether a computer file, magnetic tape, or the corner of a napkin. In this regard, it is irrelevant whether the capture takes the form of source[16] code (legible to humans) or a binary form[17] (an executable program); both forms benefit from the protection. Conversely, as long as it remains in the mind of its inventor, the software falls outside of the purview of the *Copyright Act* owing to the lack of a record of the idea on a tangible medium.[18] Furthermore, ownership of this (tangible) medium should not be confused with ownership of the (intangible) work. While the purchaser of a CD-ROM obtains ownership of the disk, by default

---

[13] *The Constitution Act*, 1867, section 91(23), source: <http://www.canlii.org/ca/const_en/const1867.html>.

[14] *Copyright Act*, *op. cit.*, footnote 10, sect. 2.

[15] *CCH Canadian Ltd.* v. *Law Society of Upper Canada*, 2004, SCC 13, Section 24, source: http://www.canlii.org/ca/cas/scc/2004/2004scc13.html.

[16] *IBM Corp.* vs. *Ordinateurs Spirales*, [1985] 1 C.F. 190.

[17] *Apple Computer* vs. *Mackintosh Computers Ltd.*, [1987] 1 C.F. 173.

[18] Laurent CARRIÈRE, "Protection du logiciel et autres oeuvres originales en vertu de la Loi sur le droit d'auteur et conventions usuelles en ce domaine" (1996) *Léger Robic Richard*, source : <http://www.robic.ca/PublicationsFr.shtml>.

this does not confer any claim of copyright on the material contained on that disk on him.

[22] To the extent that software meets the conditions described above, the copyright regime automatically applies, without further formalities. Absolutely no registration or disclaimer is required: copyright exists as of the creation of the work. That does not, however, imply that these operations are valueless. They both serve to facilitate proving the existence of a copyright and establishing its ownership.[19] Consequently, addition of a header containing the author's name and a copyright disclaimer to the source code of the software prevents third parties having downloaded it from pleading ignorance.

[23] A final characteristic of copyright is that it is infinitely divisible. Thus, it is incumbent on the copyright owner to determine who can use it, how, for how long, where, etc. The law makes provision for two types of transactions involving the protection afforded by copyright: assignment and grant.

[24] Assignment involves transferring ownership of the rights over the software. Once this has occurred, the initial owner has forfeited all claims, which are now vested in the assignee. Thus, for example, assignment allows the owner of the copyright to shift responsibility for the exploitation of the software to a third party. However, in this situation, the grant proves a more appropriate approach. The grant, expressed as a licence, makes it possible for the third party to perform certain acts which would otherwise violate the copyright, without any transfer of ownership. In brief, assignment is to sale as grant is to lease. Owing to these particularities, software is nearly always supplied to end users under a licence.

[25] Furthermore, this type of licence has the effect of creating a contractual relationship between the copyright owner and the end user. On one side, the licensor expresses his intention to grant some part of his exclusive rights over the software, while on the other side the licensee expresses his consent to use the software in compliance with the terms specified in the licence. This consent may be given in a variety of forms: by breaking the packaging of a box, by clicking an on-screen button, or simply by using the software. The primary consequence following from this convention is that the licence is covered by contract law.[20]

---

[19] *Copyright Act*, op.cit., footnote 7, sec. 34(4) and 53(2).
[20] David VAVER, *Copyright Law*, Irwin Law, Toronto, 2000, p. 229.

[26] Since the legal rules governing contracts fall under private law, Québec civil law should be applied to software licences in Québec. Thus, provisions of the CCQ governing obligations supplement the *Copyright Act* in all matters related to contractual liability of the licensor and the licensee. Similarly, every Québec law containing provisions affecting contractual relationships is applicable. This proves particularly useful in light of the lack of precision of the *Copyright Act* in this matter.

[27] Since freedom of the parties is the guiding principle in contractual matters, it is not surprising that software licences are characterized by a wide diversity.

5.2. The different types of software licences

[28] Since copyright law vests all the rights over software in its author and allows him to divide these rights as he sees fit, there is considerable flexibility in how software licences can be drawn up. Consequently, the number of licences in existence is virtually unlimited—some are extremely restrictive, while others are just as liberal. It is, nonetheless, possible to group them into a few categories that quite accurately reflect the context in which software is usually made available to users.

[29] Most of the software available today is proprietary, i.e. a firm or individual reserves exclusive rights over its exploitation. Proprietary licences confer only a usage right on the licensee, and graft a number of restrictions onto it.[21] These restrictions forbid modifying, copying, and redistributing the received copy of the software. Other limitations may affect the warranty provided, issues of liability, or the choice of jurisdiction, for example. To ensure the effectiveness of these restrictions, proprietary software is distributed in the form of binary executables that are illegible to human beings.

[30] Proprietary software may use various distribution models over the Internet. For example, the distribution of freeware is usually allowed because, it is freely distributed from the beginning. However, its use is circumscribed and its modification is forbidden. Similarly, shareware can be redistributed and used for free during a limited period. However, at the end of that period the licensee must pay a fee to the licensor to get the right to continue using the software.

[31] At the other end of the spectrum, some licences have the effect of putting the software into the public domain. This means that the authors relinquish all

[21] William H. NEUKOM and Robert W. GOMULKIEWICZ, "Licensing Rights to Computer Software", (1993) 354 *PLI/Pat* 775.

protections afforded them by copyright law.[22] Under this type of licence, the owner of the rights does not retain any particular claim on the software, which can subsequently be used, modified, and redistributed without restriction.

[32] Free and open source licences are at the midpoint between these two extremes. Under the criteria established by the FSF, software is considered free if its licence allows users the following four fundamental freedoms:

- The freedom to run the program, for any purpose.
- The freedom to study how the program works, and adapt it to their own needs.
- The freedom to redistribute copies of the program.
- The freedom to improve the program, and release their improvements to the public.[23]

[33] The effectiveness of these freedoms first requires that neither permission nor payment may be required before they can be exercised. Parenthetically, the cost of the software must not include any compensation for copyright, in contrast to firmly established practice in the case of proprietary software. Only the reproduction and distribution cost should be included in the price charged for the software.

[34] Second, the source code must be made available for the licensee to be able to study and modify it. It must be attached to the software or available on demand. Access to the source code is, indeed, the determinant characteristic of free and open source software.

[35] Third, for these freedoms to be meaningful, they must be irrevocable. Thus, if the licensor reserves the right to unilaterally withdraw the rights granted, the software cannot be considered free and open source.

[36] Under these terms, free and open source licences authorize the users of the software to perform most of the actions that are normally precluded by copyright law, while ensuring that ownership of the copyright remains in the hands of the licensor. Among other things, this allows the latter to impose various types of conditions on the use of the software. The most well-known of these conditions is

[22]Jean-Paul SMETS-SOLANES and Benoît FAUCON, *Logiciels Libres: Liberté, Egalite, Business, Freepatents*, Edispher, 1999.

[23]FREE SOFTWARE FOUNDATION, "The Free Software Definition", (2005) *GNU*, source: <http://www.gnu.org/philosophy/free-sw.html>.

undoubtedly the one included in the GPL that obliges licensees who wish to distribute a modified version of the software to do so under the same licence.

[37] Finally, some licensors, while attracted to the principles underlying free and open source software, are not prepared to bear all the consequences that result from the usage of a licence of this type. This reluctance has given rise to a variety of licences called "new" or "semi" free. These licences are much more permissive than proprietary licences, and they all provide for making the source code available, but they restrict the fundamental freedoms associated with free and open source software in some fashion.

[38] In certain cases, the software licence may grant the right to use, redistribute, and modify the software, but stipulate that this is limited to non-profit users or require the payment of a fee for commercial applications. Under these public source licences, third parties are prevented from realizing a profit from the use or distribution of free and open source software, while the copyright owner is not. An example of this is the Aladdin Free Public License.[24] This type of licence violates the first of the freedoms associated with free and open source software, which is the condition of non-discrimination as to the use of the software.

[39] Sometimes the licence will allow the source code to be modified, but under the condition that any implementation of this modified code adheres to certain standards established by the licensor. This is essentially equivalent to subjecting the modification and its distribution to his approval. This approach has been taken by, among others, Sun Microsystems with its Sun Community Source License.[25]

[40] To satisfy demands from its customers, even Microsoft has developed a semi-free licence, called a Shared Source License.[26] Its main feature is to allow licensees to study a part of the code developed by Microsoft. However, modification and redistribution of this code is severely limited.

[41] Since these various solutions derive from free and open source software, it is likely that the legal framework applicable to the latter will extrapolate to the

[24]ALADDIN ENTERPRISES, "Aladdin Free Public License", (2000) *University of Wisconsin*, source: <http://www.cs.wisc.edu/~ghost/doc/cvs/Public.htm>.

[25]SUN MICROSYSTEMS, "Sun Community Source Licensing (SCSL) - Principles", (2005) *Sun*, source: <http://www.sun.com/software/communitysource/principles.xml>.

[26]MICROSOFT CORPORATION, "Microsoft Shared Source License", (2005) *Microsoft*, source: <http://www.microsoft.com/resources/sharedsource/default.mspx>.

former…at least partially. However, this statement will need to be validated for the specific terms of each licence.

[42] Moreover, the flexibility available to software copyright owners does not end there. In fact, since each licence constitutes an individual grant, nothing stops the licensor from granting different rights to various licensees.[27] Thus, it is possible for the same software to be subject to different licences, which sometimes allows users to choose the one that best suits their needs. This is the case, for example, with the MySQL database, which is distributed under both the GPL and a proprietary licence.[28] This allows MySQL AB to distribute its software for free on the Web while selling it to clients who are interested in incorporating it into their own, proprietary software.

[43] Similarly, the way in which software is licensed can potentially change over time. This can occur either by providing for an evolution in the rights granted within a single licence, or by distributing software on different terms at various points in time. Ghostscript is an example of software that has been marketed in such a way. One year following its release, the Ghostscript code becomes available under the GPL, but users seeking to develop commercial applications on the basis of more recent versions must accept the terms of a proprietary licence.[29]

[44] In a word, the distribution of software is characterized by the wide diversity of contractual arrangements governing it. Furthermore, simply making the source code available to the users does not suffice to impart the fundamental freedoms advocated by the defenders of free and open source software. In fact, even these latter do not always agree on the exact nature of the software they promote.

5.3. Free licences vs. open source licences

[45] In 1998, a split occurred in the free software community.[30] Some programmers, primarily under the influence of Éric Raymond and Bruce Perens, began to distance themselves from the ideology of the FSF, deeming it poorly adapted to today's business climate. According to them, the term "free software" was unlikely to motivate the software industry to adopt their development model,

---

[27]Andrew M. ST. LAURENT, *Understanding Open Source & Free Software Licensing*, O'Reilly, Sebastopol, 2004, p. 162.

[28]MYSQL AB, "MySQL Commercial License", (2005) *MySQL*, source: <http://www.mysql.com/company/legal/licensing/commercial-license.html>.

[29]ARTIFEX SOFTWARE, "Licensing info...", (2005) *Artifex*, source: <http://www.artifex.com/licensing/>.

[30]OPEN SOURCE INITIATIVE, "History of the OSI", (1999) *Opensource*, source: <http://www.opensource.org/docs/history.html>.

since it implied a predominance of ethical and moral ideals.[31] Consequently, they adopted a new strategy based on the notion of open source software.

[46] Like the FSF definition, that of the Open Source Initiative (OSI) protects the right to use, redistribute, access the source code of, and modify software.[32] Indeed, on a theoretical level, free software and open source software are identical. Only the terminology was changed to switch the emphasis from the freedom of the user to the availability of the source code. Ultimately, the distinction is essentially philosophical: The open source model emphasizes pragmatism, while the free software model is based on ethics.[33]

[47] However, on a practical level, some differences between the two camps eventually appeared. These mostly involved firms that provide software and give access to the code while denying some of the other essential freedoms to the user. These actions were sometimes accepted by the advocates of open source, but are categorically rejected by free software supporters. This was the case, for example, with the initial version of the Apple Public Source License[34] (APSL) used by Apple for its Mac OS X operating system. This licence, in addition to compelling the licensees to publish any version of the computer program that they had deployed, and to notify Apple thereof, also included a provision under which Apple could revoke the licence in the event of copyright or patent litigation. These restrictions on the licensees' freedoms were compatible with the terms of the OSI, but not with those of the FSF. This debate was put to an end with the changes made to the APSL in version 2.0.[35]

[48] There is no legal distinction between free software and open source software. To understand this, it is sufficient to observe that the two movements are based on the same ideology and the same legal documents. Therefore, we use the term "free and open source" throughout this study to describe the software under examination.

---

[31]Sam WILLIAMS, Free as in Freedom: Richard Stallman's Crusade for Free Software, O'Reilly, Sebastopol, 2002, p. 166.

[32]OPEN SOURCE INITIATIVE, "The Open Source Definition", (2005) *Opensource*, source: <http://www.opensource.org/docs/definition.html>.

[33]Brett WATSON, "Philosophies of Free Software and Intellectual Property", (1999) *RAM*, source: <http://www.ram.org/ramblings/philosophy/fmp/free-software-philosophy.html>.

[34]APPLE COMPUTER, "Apple Public Source License (Version 1.2)", (2001) *Apple*, source: <http://www.opensource.apple.com/apsl/1.2.txt>.

[35]APPLE COMPUTER, "Apple Public Source License (Version 2.0)", (2001) *Apple*, source: <http://www.opensource.apple.com/apsl/2.0.txt>.

[49] Despite this uniformity in terms of the applicable legal framework, free and open source licences are not all fully equivalent. While some simply consist of a few paragraphs, others are very elaborate. It is, however, possible to classify them all into two main groups, depending on whether or not they make use of the copyleft mechanism.

5.4. Non-copylefted free and open source licences

[50] Non-copylefted licences include most of the free and open source licences that were designed within educational institutions. The majority of these were written by academics seeking to make software developed by their schools available to the public while imposing the smallest possible number of obligations on the licensees. These licences, for all their simplicity, cover a broad array of software that makes up the infrastructure of the Internet.

[51] The legal status of non-copyleft free and open source licences is similar to that of public domain software. As a result, some authors refer to them as "public-domain type" licences,[36] because of the breadth of rights they grant. In fact, they generally authorize the use, modification, and redistribution of the source code with no additional restrictions. However, software that is distributed under these licences is not in the public domain, since they specify that the owner of the copyright does not surrender it. Moreover, they often contain an obligation to reveal the existence of this copyright within the modified versions of the software.

[52] Unlike copylefted software, non-copylefted software allows source code subject to it to be incorporated into software governed by another licence. Any licensee may thus modify the software and redistribute it under the licence of his choice, including a proprietary one.

[53] Among all free non-copylefted licences, the Berkeley Software Distribution (BSD) licence is unquestionably the best known.[37] Among other software, this licence covers the various implementations of the BSD Unix operating system. This licence is non-restrictive, allowing more or less anybody to do just about anything with the code covered, provided that the licensee agrees to:

- Mention the existence of a copyright on the source code;

[36] J.-P. SMETS-SOLANES and B. FAUCON, *op. cit.*, footnote **Erreur ! Signet non défini.**.
[37] OPEN SOURCE INITIATIVE, "The BSD License", (2005) *Opensource*, source: <http://www.opensource.org/licenses/bsd-license.php>.

- Mention the existence of a copyright on the binary code and the documentation;
- Refrain from using the licensor's name to endorse or promote modified versions of the software without permission;
- Recognize that the software is provided without warranty.

[54] Until 1999, the BSD licence also included an advertising clause, requiring the licensee to mention the developers' names in any promotional material linked to the software. After many years of development, this clause engendered a perverse result: The mandatory credits in some software contained dozens, if not hundreds, of names.[38] Consequently, management of these lists of names eventually became a barrier to the freedom to modify the software, which is in contradiction with the spirit of free and open source software. For this reason, the licence was modified to suppress the advertising clause.

[55] The Massachusetts Institute of Technology (MIT) also created its own free and open source licence.[39] It applies, among others, to the X windowing software. Since the MIT licence was based on the BSD licence, the two became practically identical after the advertising clause was suppressed in the latter. However, the MIT licence is slightly more permissive, as it does not contain a non-endorsement clause for modified versions of the software.

[56] The Apache[40] licence is certainly the most elaborate of all the non-copylefted licences. This is even more true since the publication of version 2.0, early in 2004. This licence, developed by the Apache Software Foundation (ASF) for its celebrated Web server, is increasingly prevalent in academic institutions.

[57] In addition to the terms imposed by the BSD licence, the Apache licence specifies that modifying or distributing the product requires:

- An indication in each modified file that it was modified by the licensee;
- Inclusion of the disclaimer in the licence whenever a modified version of the software is redistributed.

---

[38]FREE SOFTWARE FOUNDATION, "The BSD License Problem", (2004) *GNU*, source: <http://www.gnu.org/philosophy/bsd.html>.

[39]OPEN SOURCE INITIATIVE, "The MIT License", (2005) *Opensource*, source: <http://www.opensource.org/licenses/mit-license.php>.

[40]OPEN SOURCE INITIATIVE, "Apache License", (2004) *Opensource*, source: <http://www.opensource.org/licenses/apache2.0.php>.

[58] Furthermore, this licence contains a few other provisions that are absent from other non-copylefted free and open source licences. In particular, it takes into account the other regimes of intellectual property, providing for the grant of any patent necessary for the exercise of the other rights ceded, and specifically denying any right to use the licensor's trademarks. It also explicitly authorizes the licensee to release modified versions of the software under another licence, provided that the terms of the Apache 2.0 licence are not violated. However, the most important of these additional clauses is the one specifying the regime applicable to the contributions made by licensees for purposes of having them integrated into the official software. If no other licence is designated, the provision states that these contributions are automatically subject to the Apache 2.0 licence. The advantage of this stipulation is that it guarantees the project promoter's ownership of all the rights required to re-license the official software, including the various contributions.

[59] The Academic Free License[41] and Artistic License[42] are other common non-copylefted free and open source licences. These two, like several dozen others, are derivatives of the BSD or Apache licences. The main purpose of all of these licences is to promote the dissemination of knowledge by imposing the fewest possible conditions on the use, modification, and redistribution of software. Other licences, using the technique of the copyleft, seek to pursue a similar objective by imposing additional conditions on licensees.

5.5. Copylefted free and open source licences

[60] The term copyleft was proposed by the FSF as an alternative to copyright. The purpose of the copyleft is to use the protections provided by copyright to guarantee freedom of use and modification for software.

[61] First and foremost, copylefted licences provide a heightened level of freedom to the licensee by allowing him to use, modify, and redistribute the software. Their unique feature, however, is that they extend the same freedoms to the licensees of all software based on any modification to the code of the original software.[43] This is accomplished by the inclusion of a clause that stops the licensee from distributing a

[41]OPEN SOURCE INITIATIVE, "Academic Free License, v. 2.1", (2005) *Opensource*, source: <http://www.opensource.org/licenses/afl-2.1.php>.

[42]OPEN SOURCE INITIATIVE, "Artistic License", (2005) *Opensource*, source: <http://www.opensource.org/licenses/artistic-license.php>.

[43]FREE SOFTWARE FOUNDATION, "What is Copyleft?", (2005) *GNU*, source: <http://www.gnu.org/copyleft/copyleft.html>.

modified version of the software under more restrictive terms. Consequently, incorporating copylefted computer code into proprietary software is not allowed.

[62] The first software licence to integrate the copyleft mechanism was the GPL. To this day, it is still the most widely known and commonly used free and open source licence. It continues to govern the majority of free and open source software, including the operating system GNU/Linux. It confers upon licensees the various rights promoted by the FSF, on condition that they agree to:

- Refrain from asserting copyright over the software;
- Supply the source code to all those who obtain the binary code;
- Insert a notice into the software specifying that the GPL applies to it;
- Recognize that the software is provided without warranty;
- Distribute all modified versions under the same terms.

[63] It is the commitments expected from the licensee that have led certain authors to speak of the "reciprocity" of the GPL.[44] The principal impact of this reciprocity is a constant expansion in the amount of code subject to the GPL. Indeed, each time external code is integrated into code protected by the GPL, or vice versa, the result must be distributed under the GPL. Detractors of this licence criticize this regimentation, which makes it impossible for individuals or firms unable to accept the other conditions of the GPL to benefit from all this freely available code. Conversely, those who do use the GPL respond by observing that it is precisely that mechanism that safeguards the fundamental freedoms they advocate.

[64] At the very least, the case of libraries presents real difficulties when they are subject to the GPL. The role of libraries is to be linked to other programs so as to create executable software. Unfortunately, it is possible to interpret the GPL to mean that any software relying on the use of a library that is governed by this licence must also be distributed under the same licence. However, in this event, very few developers of proprietary software would be inclined to use copylefted libraries.[45] To address this issue, the FSF released a more flexible licence, the GNU Lesser General

[44]Lawrence ROSEN, Open Source Licensing: Software Freedom and Intellectual Property Law, Prentice Hall PTR, Upper Saddle River, 2004, p. 103.

[45]Dennis M. KENNEDY, "A Primer on Open Source Licensing Legal Issues: Copyright, Copyleft and Copyfuture", (2001) 20 *St. Louis U. Pub. L. Rev.* 345, 362.

Public License[46] (LGPL). The primary purpose of the LGPL is to provide that, under some circumstances, software resulting from the use of an LGPL library is not considered a modified version of that library.

[65] The Mozilla Public License[47] (MPL) is another copylefted free and open source licence. Designed as a less restrictive implementation of the copyleft mechanism, it is essentially a compromise between the BSD and GPL licences. While it includes most of the provisions of the GPL governing modified versions and redistribution, it does allow code subject to it to be integrated into a larger project that is under a different licence, such as the BSD. In this case, only the portion that is a modified version of code initially covered must be released under the terms of the MPL, while the rest of the project could be under a proprietary licence. In any event, litigation involving interpretation of the official version of the MPL is not liable to fall under Québec law, since it contains a clause specifying that California law applies.

[66] Once again, this handful of licences has given rise to a plethora of derived licences. Thus, any difficulty arising from their application requires scrutiny of the specific terms of the licence in question. Nonetheless, it is possible to identify a certain set of legal issues that are common to all free and open source licences, the solution to which may vary depending on whether or not they are copylefted.

[46]FREE SOFTWARE FOUNDATION, "GNU Lesser General Public License", (2005) *FSF*, source: <http://www.fsf.org/licensing/licenses/lgpl.html>.

[47]Mozilla Organization, "Mozilla Public License Version 1.1", (2005) Mozilla, source: <http://www.mozilla.org/MPL/MPL-1.1.html>.

6. The legal validity of free and open source licences

[67] The nature and content of free and open source software licences are unique. Some authors even assimilate copyleft licences into viral contracts, since these automatically accompany the digital code. On account of these particularities, detractors of free and open source software contest the validity of the licences, claiming that they cannot be considered legally binding. This question is fundamental, since their validity ensures the legal security required to envisage their use. While the law continues to lag in directly addressing free and open source licences, a certain international jurisprudence has begun to emerge having pronounced in favour of the validity of these licences (though it remains embryonic). Holding them up to Québec law suggests a similar outcome. Nonetheless, no legal proceeding, whether in Québec or elsewhere, has definitively ruled on this issue.

6.1. The discussion on the international level

[68] In the United States, some authors question the viability of the FSF philosophy, claiming it violates the principles promulgated by copyright. According to this conception, the validity of a licence that uses the copyright regime to eliminate intellectual property rights over derived works is open to challenge.[48] This argument is usually based on Section 8, Clause 8, of the U.S. constitution, which identifies the purpose of intellectual property, namely to "promote the Progress of Science and useful Arts." However, this argument is flawed, is as much as free and open source software has, in fact, been at the root of many technological innovations having seen the light of day in recent years.

[69] Taking the argument a little further, some claim that copyleft licences are simply informal restrictions encumbering the information, and that they are not executable as long as the licensee has not directly contracted with each of the many preceding licensors.[49] Eben Moglen, legal counsel to the FSF, responds to this:

> "There's no absence of privity [which isn't required anyway]. […] In the case of the GPL, no one is bound to anything in particular unless she redistributes the software, modified or unmodified. Because copying and redistribution, or the making of derivatives, are never authorized in the

[48]Mark LEMLEY, Peter MENELL, Robert MERGE and Pamela SAMUELSON, *Software and Internet Law*, New York, Aspen Publisher, 2001, p. 532ff.

[49]Robert P. MERGE, "The End of Friction? Property Rights and Contract in the 'Newtonian' World of On-Line Commerce", (1997) 12 *Berkeley Tech. L. J.* 115.

absence of a license, undertaking to redistribute is clear acceptance of our terms for redistribution"[50].

[70] Indeed, free and open source licences generally appear to comply with legal requirements, at least to the extent that the licensee's consent can be ascertained.[51] It is, in fact, the assumption that the terms of the licence have been accepted at such a time as when the software is redistributed that raises concerns regarding the opposability of the contract. Indeed, if the user is unaware of the implications of this action, it is not clear that privity exists between the parties. Consequently, each situation represents a particular case subject to the discretion of the courts. As a general rule, for the terms of the licence to be enforceable against the licensee, this latter must have paid sufficient heed to the clauses.

[71] Most commonly, free and open source licences are attached to the software they encumber. This is how they can be associated with the regime of shrinkwrap, clickwrap, and browse-wrap licences, all of which provide similar methods for obtaining the consent of the licensee. There is no doubt today that U.S. law accepts the legitimacy of shrinkwrap and clickwrap licences—the ProCD, Inc v. Zeidenberg[52] decision provides the reference. However, U.S. courts generally refuse to recognize application of browse-wrap type licences, i.e. those posted on a Web site, especially when the user has not accepted or viewed the document before downloading the software.[53]

[72] The issue of consent, taken a little further, also raises questions regarding the capacity of the free and open source software licensee to sub-license the software in turn. In fact, it is not clear that he owns the rights and can consequently grant a license himself. It is the absence of proximity between the developers that creates uncertainty regarding ownership of the rights governing the modified software, to which several individuals will have contributed without, necessarily, forfeiting their own rights. This absence of "privity," as the term is used in U.S. law, i.e. the contractual relationship between all parties, can raise doubts regarding the capacity of any one of them to sub-license software over which he does not own all

[50]Eben MOGLEN, quoted in Denis E. POWELL, " Judgment Day for the GPL? ", (2000) *LinuxPlanet*, source : <http://www.linuxplanet.com/linuxplanet/reports/2000/1>.

[51]Mélanie CLÉMENT-FONTAINE, "La licence publique générale GNU", (1999) *Juriscom*, par. 32. Source: <http://www.juriscom.net/uni/mem/08/presentation.htm>.

[52]*ProCD, Inc* v. *Zeidenberg*, 86 F.3d 1447 (7e Cir. 1996).

[53]*Specht* v. *Netscape Communications Corp.*, 150 F. Supp. 2d 585 (SDNY, 2001).

intellectual property rights. In this scenario, the licence may be void or, at least, non-binding, precisely because of this lack of proximity.

[73] There is another characteristic of free and open source licences that raises doubts in the minds of some authors regarding the validity of these contracts under U.S. law. These are the "grantback" clauses present in copyleft licences. These licences extend the right to use, modify, and redistribute to the licensee, provided that he transfers his copyright over the derived work based on the licensed software in advance. In U.S. law, these provisions could constitute a case of "copyright misuse," which would strip the contract of its binding effect.[54]

[74] Despite these many challenges raised by the U.S. literature, it remains the case that, for the most part, free and open source licences are drawn up in consideration of the law applicable in the United States. With respect to foreign laws, some of their clauses may be inapplicable or void if they conflict with a local public policy rule. This may have the effect of invalidating either the clause in question or the entire licence.

[75] Thus, several clauses found in most free and open source licences are deemed inapplicable in France. The principal difficulty arises because of their incompatibility with the provisions of French law governing intellectual property rights. This is particularly the case with respect to the formalism requirements for assigning and granting rights. French intellectual property rights law requires that "the domain of the rights granted be limited in terms of their scope, destination, location, and duration."[55] (authors' translation) Since free and open source licences almost never contain such limitations on the scope of the exploitation of the rights, they are inevitably void. As a consequence, a licensee redistributing the software is committing an unauthorized act. Moreover, the clause under which the licensee renounces his proprietary rights over the derived software in advance also creates a problem under French law. In this matter, the law prohibits any renunciation of future rights[56] and any global assignment of future creations.[57] In any event, non-compliance with the aforementioned formalism implies that assignment in advance is invalid, and consequently all subsequent licences are invalid.

---

[54]Christian H. NADAN, "Open Source Licensing Virus or Virtue" (2002) 10 *Tex. Intell. Prop. L.J.* 349.

[55]*Code de la propriété intellectuelle*, section L. 131-3.

[56]J. Carbonnier, *Droit civil, Introduction*, PUF, 25th edition, 1997, No. 185.

[57]*Code de la propriété intellectuelle*, section L. 131-1.

[76] Aside from these key incompatibilities between free and open source licences and French intellectual property rights law,[58] many other problems arise. There is, for example, violation of French consumer law, especially in matters of abusive clauses[59] and the regulation of contracts concluded remotely.[60] Nonetheless, this law applies inasmuch as the licensee assumes the guise of a consumer. These licences are also vulnerable to criticism for failing to comply with the Toubon Act regarding the use of the French language,[61] rules relative to the formation of electronic contracts,[62] and, in particular, regulations governing the use of limitation and disclaimer clauses. In fact, these clauses are prohibited by French law except under certain restrictive conditions. Notably, they must be consistent with the essential commitments of the licensor.[63]

[77] It was this proliferation of obstacles that motivated CEA, CNRS, and INRIA to launch a project to draw up a French free and open source licence, so as to ensure greater legal protection while preserving the spirit of these licences as much as possible.[64] This project gave rise to the CeCILL licence in July of 2004. While French authorities welcomed this initiative,[65] the practical impact of the introduction of such a licence with a national flavour remains to be established. In any event, there is some reason for concern in that, by trying too hard to comply with domestic laws, this type of licence fails to account for the international character of the free and open source software development model. The geographical distribution of developers will result in a large proportion of the free and open source software used in France being subject to the GPL. Moreover, it will not be possible to integrate the source code of those that are governed by CeCILL into many existing free and open source projects, owing to incompatibilities between CeCILL and the GPL.

---

[58]Christophe CARON, "Les licences de logiciel dits 'libres' à l'épreuve du droit d'auteur français", (2003) 23 *Dalloz* 1556.

[59]*Code de la consommation*, section L. 132-1 and following.

[60]*Code de la consommation*, section L. 121-16 and following.

[61]V. M-L. DE CORDOVEZ, S. LIPOVETSKY, "La loi Toubon confrontée aux nouvelles technologies de l'information", (2002) June CCE 16; A. LEPAGE, "Libertés et droits fondamentaux à l'épreuve de l'internet", Litec, 2002, No. 196 s.

[62]Code de la consommation, sect. L. 121-16 and following; Act *2004-575 of June, 21, 2004 "pour la confiance dans l'économie numérique"*, J.O. No. 143 of June 22, 2004, p. 11168.

[63]C. Cass. Com., October 22, 1996, Bull No. 261 (Arrêt Chronopost).

[64]CEA, CNRS, INRIA, *loc. cit.*, footnote 8.

[65]Press release: "Renaud DUTREIL soutient CeCILL, la première licence française de logiciel libre élaborée par le CEA, le CNRS et l'INRIA", July 5, 2004. Source: <http://www.fonction-publique.gouv.fr/leministre/lescommuniques/communique-200407061150.htm>.

[78] Internationally, the conflict between SCO and several software publishers, including Novell, Red Hat and, especially, IBM, has been a particular focus in terms of the validity of free and open source licences. This major court case, currently before the United States District Court for the District of Utah, has dragged on since March of 2003. SCO accuses IBM, under colour of breach of contract and revealing trade secrets, of having introduced parts of the Unix code over which SCO possessed intellectual property rights into the Linux kernel[66]. This matter could cause the judges in Salt Lake City to examine the validity of the GPL and LGPL licenses, partly because of IBM's counterclaims, but mostly because of SCO's claims that they are unconstitutional. SCO maintains that the GPL, under which Linux is licensed, violates the constitution of the United States as well as U.S. copyright and patent laws.[67] According to SCO, the copyleft system is equivalent to a total negation of the relevant law and is contrary to the objectives and the spirit of the texts referred to, which forcefully protect products of the intellect. However, the judges have not yet examined SCO's claims and no core decision is expected before the autumn of 2005.

[79] In fact, there are few legal decisions, globally, that either countenance or repudiate the validity of free and open source software licences. Very often conflicts in this matter are settled by means of negotiation and out-of-court settlements between the parties.[68] Indeed, software developers generally prefer to withdraw the offending code or submit the software in its entirety to the licence, rather than be liable for damages plus interest or fines for copyright infringement, or to incur the opprobrium of a large segment of their community.[69] An additional reason that can be invoked in this context concerns the self-executing character of free and open source licences. In fact, the burden of the obligations placed on licensees is so light that they cannot generally benefit from challenging the validity of the contractual agreement binding them to the copyright owner.[70] When they do so, they place themselves under generic intellectual property rights rules, which are much more stringent.

---

[66]Kerry D GOETTSCH, "Sco Group v. IBM: the future of open-source software" (2003) *U. Ill. J. L. Tech. & Pol'y* 581.

[67]Darl MCBRIDE, "Open Letter on Copyrights", (2003) *SCO*. Source: <http://www.sco.com/copyright/>.

[68]These bilateral transactions are, by their very nature, confidential. See, for example, settlements between Netfilter and Fujitsu-Siemens, Asus, and Securepoint. Source: <http://www.netfilter.org/news>.

[69]Mark H WEBBINK, "Open Source Software – Bridging the Chasm", (2002) 691 *PLI/Pat* 663, 683.

[70]A. ST. LAURENT, *op. cit.*, footnote **Erreur ! Signet non défini.**, p. 151.

[80] Two legal decisions relative to the validity of the GPL licence can nonetheless be identified.

[81] The first is a decision, in the form of a preliminary injunction rendered by the Federal Court in the U.S. state of Massachusetts in the case of MySQL v. Progress Software.[71] This lawsuit deals with Progress Software's proprietary Gemini product, which dynamically links to the free and open source database software MySQL. Since the source code of Gemini was not freely available, MySQL demanded that its distribution be suspended. While the judge did not find for this demand, in the absence of proof of harm, she implicitly considered the GPL to be a binding licence. However, as an out-of-court settlement was ultimately reached in this case, no core decision was every rendered to confirm this preliminary ruling.[72]

[82] A second decision, which also took the form of a preliminary injunction, yielded a more unambiguous ruling on the validity of free and open source licences. This decision, from the Regional Court of Munich, Germany, dated April 2, 2004, in the Netfilter/iptables v. Sitecom Germany GmbH case,[73] was confirmed on appeal by the same court on May 19, 2004.[74] In this case, Sitecom was distributing wireless routing software based on another software product developed by Netfilter/iptables and released under the GPL. Sitecom's product was distributed with no access to the source code and without including the GPL or, for that matter, even mentioning it. The Munich court ruled that Sitecom had violated the terms of the licence and the company was enjoined to comply with it or cease distributing the software in question. This is the first judicial decision to have clearly pronounced on the validity of the GPL and to have ordered that compliance be enforced. Once again, however, the reach of this decision must be kept in perspective, since a preliminary decision does not bear the weight of a final decision on the core of the lawsuit.

[83] In summary, the global sparseness of case law in the matter of free and open source licences does not justify the conclusion that these contracts have been validated in all points by the courts. At most, we can assert that some jurisdictions have considered these licences to have a certain binding force. It is worth noting

[71] *Progress Software Corp.* v. *MySQL AB*, 195 F.Supp.2d 328; Laura A. MAJERUS, "Court Evaluates Meaning of 'Derivative Work' in an Open Source License", (2003) *Findlaw*. Source: <http://articles.corporate.findlaw.com/articles/file/00050/008924>.

[72] MYSQL, "MySQL AB and Nusphere Corporation Announce Settlement", (2002) *MySQL*. Source: <http://www.mysql.com/news-and-events/press-release/release_2002_14.html>.

[73] The original decision, in German, is at: <http://www.ifross.de/ifross_html/eVWelte.pdf>.

[74] Translation of the decision into English. Source: <http://www.jbb.de/judgment_dc_munich_gpl.pdf>.

that their validity remains conditional on compliance with the applicable law, which is highly variable in light of the international development model of this software. Consequently, it is a matter of elaborating solutions on a case by case basis, as a function of applicable national law.

[84] However, it appears promising to invoke custom to validate the legal mechanism chosen by free and open source licences, even when their legal bases are weak or contested. This is what some advocate by asserting:

> "The positive legal framework on which licensing depends might have shaky conceptual foundations, but it might be supportable, nonetheless, if its historical and customary pedigree is sufficiently robust. The standard software licensing model might represent an enforceable legal form simply because licensing has become the customary form of dealing in computer software"[75].

[85] In the matter of software development, it appears difficult, to say the least, to deny that a practice of using free and open source licences has become entrenched. Consequently, in a nod to history, the newest software could eventually be legally vindicated by one of the oldest foundations of law.

6.2. Québec law

[86] Notwithstanding all the arguments developed by foreign authorities around the issue of the validity of free and open source licences, it remains the case that, for them to be binding within Québec, these licences must generally comply with the law applicable in Québec. For this to obtain, they must first be compatible with the provisions of the federal Copyright Act. Inasmuch as they are, the contractual relationship established between the licensor and the licensee must also meet the requirements of Québec civil law. In light of the international aspect of the development of much free and open source software, the subordination of their licences to Québec law requires some preliminary particulars.

Application of Québec law

[87] Section 5 of the Copyright Act provides that, within Canada, copyright covers all original works whose author was, at the time of their creation, a citizen, subject, or habitual resident of a country having signed the Berne Convention. Similarly, all works that were first presented to the public within a signatory country benefit from

[75]Michael J. MADISON, "Reconstructing the Software License", (2003) 35 *Loy. U. Chi. L.J.* 275.

copyright. In practice, this means that Canadian law protects all, or substantially all, authors of free and open source software within Canada.

[88] However, this protection does not automatically extend to the application of Québec contract law to the contractual relationship established by the licence. For this to be the case, some connecting factor must exist between it and Québec jurisdiction. Designation of Québec law in the licence is the preferred connecting factor.[76] In the absence of such a clause, Québec law finds application only if it features the most substantial connection with the licence. This will be the case if the residence or place of business of the "party who is to perform the prestation which is characteristic of the act" is located within the territory of Québec.[77] Typically, Québec civil law will thus not be applicable unless the licensor resides within Québec. This solution is reasonable, inasmuch as the principal obligation issuing from a free and open source software licence, as it pertains to the licensor, consists of delivering the software to the licensee[78]. However, one important exception covers consumers, who must in all events fully benefit from protections provided by Québec law.[79]

[89] For example, use of the database software MySQL, following its download from the official site of MySQL AB site by an civil servant, will be subject to application of Swedish law under the rules of private international law applicable in Québec. Conversely, the same use, if it is by an individual on a personal Web page, will be subject to the jurisdiction of Québec law.

[90] Nonetheless, in the event of litigation, it is possible that both parties will wish to see application of Québec law, or that a judge will opt to establish closer linkage with Québec law, despite the fact that the licensor resides abroad. Indeed, execution of the terms of the licence may require intervention by the Québec justice system. Under these circumstances, application of foreign law may prove unnecessarily complex for all parties.

[91] In any event, as soon as Québec law comes into play, the compatibility of free and open source licences with the Copyright Act is at issue.

---

[76] *Québec Civil Code*, *op. cit.*, *footnote* 11, sect. 3111.

[77] *Québec Civil Code*, *op. cit.*, *footnote* 11, sect. 3113.

[78] Thibault VERBIEST, "Droit international privé et commerce électronique: état des lieux" (2001) *Juriscom*, par. 6, source : <http://www.juriscom.net/pro/2/ce20010213.htm>.

[79] *Québec Civil Code*, *op. cit.*, footnote 11, sect. 3117.

Validity under the Copyright Act

[92] First of all, in Canada there is no question of challenging free and open source licences on the basis that they are incompatible with the fundamental principles of copyright. Unlike the U.S. constitution, the Canadian constitution is mute on the powers of the Federal government vis-à-vis copyright. Moreover, the Copyright Act has neither a preamble nor a section defining its goals. In section 3 it simply provides that "…[copyright] means the sole right to produce or reproduce the work or any substantial part thereof…" It is precisely from the exercise of this right that free and open source licences derive their meaning.

[93] Furthermore, unlike in the legislation in effect in most countries of continental Europe, the Canadian Copyright Act makes no explicit provision for procedures authorizing third parties to perform acts that are prohibited by it prima facie. Consequently, in Québec there is no danger of a software licence lacking precision in terms of delimiting the domain of the exploitation of the rights granted. The licensor is perfectly free to write the licence so as to grant rights in whatever fashion he sees fit, even at this risk of being ambiguous and incomplete.

[94] Indeed, the only formality required by the Copyright Act is that the licence be in written form and signed by the owner of the right.[80] Even though this sole rule places an extremely light burden on licensors, it does create certain difficulties when confronted with free and open source licences. Indeed, since they are usually attached to the software within an electronic file, it is not clear that they universally meet the requirement for a signature.

[95] In this regard, it now seems well established that the requirement for a signature does not necessarily imply affixing a handwritten mark. A first example is provided by the Personal Information Protection and Electronic Documents Act, which specifies that an electronic signature may sometimes be equivalent to the signature required by federal laws.[81] This same law defines an electronic signature as a:

> "…signature that consists of one or more letters, characters, numbers or other symbols in digital form incorporated in, attached to or associated with an electronic document."[82]

---

[80] *Copyright Act*, *op. cit.*, footnote 10, sect. 13(4).

[81] *Personal Information Protection and Electronic Documents Act*, R.S.C. (2000), ch. 5, sect. 43. Source: <http://www.canlii.org/ca/sta/p-8.6/>.

[82] *Personal Information Protection and Electronic Documents Act*, op cit., *footnote* 81, sect. 31.

[96] However, the measures this law provides for in matters of signatures do not yet apply to the Copyright Act.[83]

[97] Québec civil law, which finds application in a suppletive manner under like circumstances, provides a second example. Indeed, the Act to establish a legal framework for information technology establishes the pre-eminence of technological neutrality in Québec, specifying that "A person's signature may be affixed to the document by means of any process that meets the requirements of article 2827 of the Civil Code."[84] The requirements in this section fall into two categories: (1) The signature must allow identification of the person affected, and (2) it must represent a manifestation of his consent. Thus, under the provisions in effect in Québec, the form of a signature affixed to a licence is not a determinant of its validity. Rather, this must be established if the method retained fulfills the inherent function of a signature.[85]

[98] In the case of free and open source licences, the general practice is for the name of the owner of the rights over the software to be included in the header or footer of the licence, as well as within each of the files comprising the software. Inasmuch as the licensor includes a disclaimer that is consistently used, the identification criteria is easily respected. As to the manifestation of consent, it is unlikely that a developer would affix his name in the proximity of a licence attached to his software without desiring compliance with the terms of the licence. Thus, it appears that this approach, which draws strongly on U.S. law, allows the formal requirements of the Copyright Act to be fulfilled.

[99] Furthermore, courts will need to show some flexibility in this matter, since jurisprudence occasionally extends to recognizing the existence of implicit or verbal licences, despite the apparent contradiction with the letter of the law. Such decisions are usually rendered when a corresponding practice or custom can be ascertained.[86] In light of the relative uniformity of free and open source licences in terms of the affixing of the licensor's name, and the fact that their use has become commonplace in the software industry, they should fall under the regime established by these

---

[83] *Canada Gazette*, Part II, vol. 138, no. 26, SOR/2004-309, sect. 1.

[84] *An Act to establish a legal framework for information technology*, R.S.Q., ch. C-1.1, sect. 39. Source: <http://www.canlii.org/qc/laws/sta/c-1.1/20051019/whole.html>.

[85] Ivan MOKANOV, "La teneur du standard de fiabilité des moyens électroniques de signature", (2004) 9 *Lex Electronica* 1, p. 36. Source: <http://www.lex-electronica.org/articles/v9-1/mokanov.htm>.

[86] *Robertson* v. *Thomson Corp.*, (2004) CanLII 32254, (ON C.A.), para. 95. Source: <http://www.canlii.org/on/cas/onca/2004/2004onca11384.html>

precedents. This is all the more true, given that jurisprudence is more flexible in matters of non-exclusive licences.[87]

[100] Assuming compliance with the formal requirements, the capacity of the licensor to grant the rights must also be evaluated. In this matter, the guiding principle is that the software's author also detains the rights.[88] Nonetheless, two exceptions can alter that state of affairs. First, the author may already have assigned ownership of the rights to a third party. Second, the Copyright Act specifies that, when an employee develops software in the exercise of his work, the employer owns the rights to it.[89] In both of these cases, a licence granted by any party other than the one who actually owns the copyright shall be void. Now, in the matter of free and open source software, contributions made by employees may easily fall into this category, especially if the work is done without the employer's knowledge. Thus, in light of this criteria, the validity of a free and open source licence must be evaluated while accounting for the context within which the affected software was developed.

[101] To the extent that the provisions of the Canadian Copyright Act do not appear to conflict with the recognition of free and open source licences within Québec, it remains that they must comply with the requirements in Québec's civil law pertaining to contractual agreements.

Validity under civil law

[102] In Québec, section 1385 of the QCC provides for the modalities of contract formation. This specifies that a simple exchange of consent between the parties suffices to create a commitment binding them, insofar as the agreement contains a clause and an object regarding public order.

[103] Consequently, the essential determinant revolves around establishing consent, in particular that of the licensee in the case of free and open source licences. Now, Québec jurisprudence is extremely flexible in matters of recognizing the manifestation of the will of the parties. This is reasonable, in light of the contractual freedom that generally obtains. Unfortunately, Québec courts have not yet had the occasion to examine the transposition of these principles into the

[87] *Les Amusements Wiltron Inc.* v. *Mainville*, [1991] R.J.Q. 1930 (C.S.), p. 1935.

[88] *Copyright Act, op. cit.*, footnote 10, sect. 13.1.

[89] *Copyright Act*, *op. cit.*, footnote 10, sect. 13.3.

electronic sphere. Under these circumstances, we can look to the decision of an Ontario court in the case Rudder v. Microsoft Corp., which recognized that a click can be a valid acceptance.[90] Thus, when the licensee must activate a multimedia acceptance procedure before being able to use or download the software, the validity of the licence is not problematic. In order to facilitate proof of consent, licensors of free and open source software would benefit from implementing this type of procedure whenever possible.

[104] However, manifestation of the licensee's will need not be express, it can also be implicit.[91] Indeed, the form of acceptance generally preferred for free and open source licences is much closer to tacit consent (the licensee is assumed to have accepted the terms, for otherwise he would be acting illegally). The licensee is thus in a situation analogous to that of a Web surfer who encounters conditions imposed on the use of a Web site by the Webmaster. Given this context, in the case of Kanitz v. Rogers Cable Inc., the Ontario Superior Court ruled that a clause posted on the site is sufficient to bind the users.[92] It is true, however, that the generality of this decision is widely questioned owing to the subjectivity of the outcome, based on the "insincere" attitude of the plaintiff.[93] Nonetheless, an important component of this decision resides in the fact that the nature of the service provided was taken into consideration by the judge in the evaluation of the disputed clause. It is thus reasonable to assume that, in the case of free and open source licences, the scope of the rights granted, along with the fact that the software is free, will tip the scales in favour of the validity of the licence.

[105] In principle, there is nothing in Québec civil law that formally repudiates the manifestation of the licensee's will in the manner favoured by those who grant free and open source licences. This is particularly true for professionals in information technology, who can no longer ignore the way these licences operate. Conversely, it is possible that considerations of fairness, or application of consumer law, will lead some court to reject this original technique for establishing consent. However, the

---

[90] *Rudder* v. *Microsoft Corp.*, (1999) O.S.C.J. Source: <http://www2.droit.umontreal.ca/cours/ecommerce/_textes/rudder.doc>.

[91] *Québec Civil Code*, *op. cit.*, footnote 11, sect. 1386.

[92] *Kanitz* v. *Rogers Cable Inc.*, (2002) 58 O.R. (3rd) 299. Source: <http://www.dww.com/decisions/kanitz_v_rogers_cable_inc.pdf>.

[93] Vincent GAUTRAIS, "Les contrats de cyberconsommation sont presque tous illégaux!" (2005). Published in the journal *Revue du Notariat*, p. 11.

likelihood of such a rejection is remote, since under those circumstances Québec judges will need to account for the benefits accruing to the licensee.[94]

[106] The cause of a licence is the parties' reasons for concluding it.[95] Thus, in the case of a Québec licensee, it is possible that a free and open source licence may be declared void if one of the parties transacted with the purpose of committing an illegal act.[96] This would be the case, for example, of a licensor who distributes software with a Trojan horse embedded in it, or a licensee who uses network administration software to gain access to a system belonging to a third party without having received authorization to do so.

[107] The object, for its part, constitutes the juridical operation envisaged by the parties.[97] The grant of copyright being specifically provided for by the law, this element of contract formation is not liable to raise issues concerning the validity of free and open source licences.

[108] Finally, since the viral aspect of copyleft free and open source licences has been invoked by some authors as a reason for their nullity, an examination of the position of Québec civil law on the matter is worthwhile. In this respect, section 1382 of the QCC recognizes the existence of "aleatory contracts," i.e. agreements under which the extent of the obligations is unknown at the time they are concluded. Similarly, section 1374 specifically states that the object of an obligation may be a future good. The fact that the licensee is making a commitment on modifications and distributions he may effect in the future precisely corresponds to this concept of uncertainty. Thus, his obligations vary in proportion to the acts undertaken. Moreover, Québec jurisprudence corroborates that, "the assignment of copyright over a future work, i.e. a work that does not yet exist, becomes lawful and enforceable as soon as the work exists."[98] While the solution will probably differ in the common law provinces, the copyleft mechanism is entirely compatible with Québec civil law.

[109] On the whole, free and open source licenses appear perfectly valid when confronted with the applicable law in Québec. Indeed, the only significant

[94] *Consumer Protection Act*, R.S.Q., 1977, ch. P-40.1, sect. 9. Source: <http://www.canlii.org/qc/laws/sta/p-40.1/20051019/whole.html>.

[95] *Québec Civil Code*, *op. cit.*, footnote 11, sect. 1410.

[96] *J. Donat Langelier Ltée* v. *Demers*, (1928) 66 C.S. 120.

[97] *Québec Civil Code*, *op. cit.*, footnote 11, sect. 1412.

[98] *Diffusion YFB Inc.* v. *Les Disques Gamma (Québec) Ltée,* (1999) C.S. 500-05-047570-997.

reservation relates to the requirement in the Copyright Act concerning the formalism of the signature. Despite everything, the interpretation liable to prevail in this matter is that their validity will be supported, in the event that this issue becomes the subject of litigation before Québec courts.

7. Intellectual property

[110] As a product of the intellect, free and open source software is primarily protected by the rules of intellectual property law. At this level, the principal role of copyright is the protection of the owner's economic rights by providing for a multitude of remedies in the event of infringement. Authorship, as well as the right to the integrity of the work, are also protected under Canadian law. These protections, which are generally well suited to the development context of proprietary software in a closed environment, create certain issues when confronted with the collaborative development process chosen by many developers of free and open source software. Thus, the proliferation of successive grants of rights requires an in-depth study of the chain of title in which the user is the final link. The specific terms of the various licences are also liable to cause problems when the time comes to combine segments of code from distinct projects. Finally, application of patent law to software, as it is practised in some countries, causes serious difficulties for the long-term viability of free and open source software.

7.1. The protection of the copyright owner

[111] The fundamental goal of copyright is to provide the owner of the rights with a monopoly over the exploitation of software or any other type of creation. It is unimportant whether the owner is the author himself, his employer, or some other individual benefiting from a right assignment. In all instances, the monopoly awarded to the owner is enforceable against any person who is subject to Canadian law. This allows developers of free and open source software to enforce the protection granted by the *Copyright Act* against third parties, i.e. against anyone having withheld consent from the terms of the licence they have chosen. Similarly, the generic rules included in the *Copyright Act* should find application in the event of a licence being invalidated.

[112] Conversely, if an agreement has been reached between the owner of the rights over the software and one of its users, then the provisions of the licence necessarily govern the relationship. An assessment of the obligations of the parties must then invoke the rules of contractual liability, as we shall describe them later. Now, the structure of all free and open source licences implies a grant of rights, provided that the licensee complies with certain obligations. In case of failure, they make no further provisions, except that the grant ceases to be in effect. Thus, even

in this situation, the protections provided by the *Copyright Act* play an important role.

[113] Copyright first protects a certain set of rights, called economic rights, which limit actions likely to have an economic impact on the owner of the rights. Some of them have particular relevance to software, including the right to execute it, to reproduce it, to translate it into another programming language,[99] to communicate it to the public, and to rent it.[100] Nonetheless, the specificity of software is such that any use thereof almost inevitably implies a reproduction. For example, installing software on a computer requires reproducing it on the hard drive of this computer. Subsequently, any reinstallation would imply another reproduction. The copyright protection afforded to software is thus greater than what prevails in the paper world. In comparison, lending a book to a friend, or selling it in a used book store,[101] does not violate any rights.

[114] However, the law does allow for two specific exceptions in which reproduction of software is allowed, regardless of whether the owner of the rights consents.[102] The first allows the owner of a copy of the software to copy it, either by adapting, modifying, or converting it, in order to ensure its compatibility with a specific computer. The second allows him to make a backup copy, which must, however, be destroyed when he ceases to own the original.

[115] Furthermore, as the Canadian law states, the right to reproduce software involves the entire package or a substantial part thereof. Consequently, any line of code taken individually is not protected and it is always possible to copy extracts of software. For instance, in the case of Delrina Corp. v. Triolet Systems Inc., a judge ruled that 60 lines of code did not constitute a significant part of a program that contains more than 14,000 lines in total, especially if rewriting them would have required approximately 20 minutes of programming time.[103] Since the source code of free and open source software is made available, it is particularly open to being copied in such a way. Evaluating the magnitude of the reproduction is thus vital to ensuring its protection. The criteria widely recognized in this matter are:

---

[99] *Prism Hospital Software Inc.* v. *Hospital Medical Records Institute et al.*, (1994) 57 C.P.R. (3d) 129 (B.C.S.C.)

[100] *Copyright Act*, *op. cit.*, footnote 10, sect. 3(1).

[101] L. LESSIG, *op. cit.*, footnote 1, p.72.

[102] *Copyright Act, op. cit.*, footnote 10, sect. 30.6.

[103] *Delrina Corp.* v. *Triolet Systems Inc.* (1993) 47 C.P.R. (3rd) 1, p. 44.

- the distinct nature of the part copied (did it require particular effort, talent, or ingenuity?);
- the degree of protection software of that type merits;
- the impact on the exploitation of the software;
- unwarranted enrichment on behalf of the party reproducing the software at the expense of its author;
- the existence of a single market on which the two software programs compete.[104]

[116] Furthermore, in the case of software, it may be wise to add a functional criteria to this list. Thus, an element that benefits from a certain degree of autonomy, whether a function or a module, for example, could benefit from protection even when it only amounts to a small proportion of the full software code.

[117] In addition to economic rights, Canadian law also recognizes the existence of certain moral rights, such as authorship and the right to integrity.[105] These rights always belong to the software author since they are unassignable, even though he may renounce them. These rights derive from continental European law and do not exist in U.S. law. Consequently, most free and open source licences were not designed to account for them. They may, however, prove to have relatively far-reaching implications for the evolution of free and open source software projects.

[118] Authorship first and foremost represents the right to claim the creation of the software, and in particular to refute any third party making such a claim falsely. Québec jurisprudence specifically recognizes the possibility of a software author to assert his authorship.[106] However, enforcement of this right must always take into account fair use in the specific context in which it takes place. In the case of free and open source software, this means the right to have one's name included in the credits along with those of the other authors, should it have been omitted. On the other hand, it would not dictate inclusion of the author's name if this is not the common practice. Indeed, past problems associated with the advertising clause in the original BSD licence have revealed the pernicious effect of excessive crediting of the authors of free and open source software. Furthermore, some free and open

[104]D. VAVER, *op. cit.*, footnote 20, p. 146.

[105]*Copyright Act*, *op. cit.*, footnote 10, sect. 14.1.

[106]*Marquis* v. *DKL Technologies Inc.,* (1989) 24 C.I.P.R. 289 (C.S.Q.).

source licences, such as the Apache licence, contain clauses that address the authorship issue. Such provisions are equivalent to contractual protection of that right, which is not recognized in all countries.

[119] The right to integrity, for its part, implies the ability to block any mutilation, distortion, modification, or use of the software that might be detrimental to the honour or reputation of the author. Consequently, all contributors to free and open source software, whether or not they have assigned their economic rights to some institution managing the project, are able to enforce a correction if their honour or reputation is impugned by subsequent developers. This could be the case, for example, with software modified by a licensee to facilitate the creation of genetically modified organisms (GMOs).[107] Furthermore, the Supreme Court of Canada appears to go even further in interpreting the right to integrity. Thus, in the Galerie d'art du Petit Champlain Inc. v. Théberge ruling,[108] the Court established that the right to integrity also covers eventual modifications to the structure of the work. According to this interpretation, the author of free and open source software always retains a right of review over modifications made by subsequent licensees, despite the permissive terms of the licence. For example, he would probably be able to block some forks in the code, which usually imply a major reorganization of the software. Application of the right to integrity to free and open source software thus violates the very spirit underlying this movement. It is, however, true that doctrine generally opposes this broad definition of the right to integrity,[109] and that this Supreme Court decision was close (4 judges to 3). Thus, it is possible to envisage a reversal of jurisprudence in this matter during the coming years.

[120] In addition, the protections provided by copyright law are not absolute. For example, it always remains legal to reproduce software for purposes of private study or research.[110] Similarly, it may be reproduced within the framework of a review or a report, provided that the author is mentioned.[111] These exceptions allow free and open source software to be circulated within a firm or to quote the code in a scientific article, for example, without necessarily including the licence under which it

---

[107]Greg R. VETTER, "The Collaborative Integrity of Open-Source Software", (2004) *Utah L. Rev.* 563, 665.

[108]*Galerie d'art du Petit Champlain Inc.* v. *Théberge*, 2002 CSC 34, source: <http://www.canlii.org/ca/cas/scc/2002/2002scc34.html>.

[109]Normand TAMARO, *The 2004 Annotated Copyright Act*, Thomson Carswell, Toronto, 2004, p. 486.

[110]*Copyright Act, op. cit.*, footnote 10, sect. 29.

[111]*Copyright Act*, *op. cit.*, footnote 10, sect. 29.1.

is released. Aside from that, recourse to the principles of fair use has few potential applications to free and open source software, given the permissive nature of their licences, which usually already authorize the acts in question.

[121] The rights granted are also limited in time. In Canada, copyright ceases to exist fifty years after the death of the author.[112] In the case of software created collaboratively, as is often the case of free and open source software, this period extends for fifty years after the death of the last author. However, when foreign authors have contributed to the software development, they have no claim to a longer protection within Canada than they would have in their home country. After this period, the software falls into the public domain and can be used by anyone without any constraints. However, owing to the disconnect between the length of the protection afforded and the average lifespan of software, these rules are not likely to ever have a real impact on free and open source software.

[122] Finally, section 27 of the *Copyright Act* specifies that performing an act that is reserved to the owner of one of the rights listed above constitutes copyright infringement. As to software, jurisprudence and doctrine have determined that infringement can be established with a five step test:

- Abstraction: separate the idea from the expression;
- Evaluation: is the software original?
- Comparison: are there enough similarities?
- Filtration: are these similarities justified?
- Quantification: do the protected similarities involve a substantial proportion of the software?[113]

[123] When this test is conclusive, the owner of the rights over the free and open source software has access to a series of remedies to enforce his copyright on the person having committed the infringement.[114] The best known of these remedies is certainly the injunction, which forces the contravener to comply with the terms of the licence or cease and desist the offending activities. Incidentally, this constitutes the basis on which the two sole decisions involving free and open source licences

[112] *Copyright Act*, *op. cit.*, footnote 10, sect. 6.

[113] Nicolas SAPP, "La contrefaçon en matière de logiciel : où en sommes-nous?", (2000) 138 *Développements récents en propriété intellectuelle* 161, p. 189.

[114] *Copyright Act*, *op. cit.*, footnote 10, sect. 34.

have been rendered to date. Remedy in damages under Québec civil law is, necessarily, much less useful in this context, since its purpose is purely compensatory. But what kind of damages can be claimed by a developer who freely distributes his software on the Internet? Similarly, the statutory damages provided for by the law are of limited use to the copyright owner for free and open source software. While section 38(1) of the *Copyright Act* allows a judge to replace damages with an amount ranging between $500 and $20,000, it is most likely that the smaller amount will be retained in light of the limited economic impact of the violation.[115] Only awarding a share of the profits earned by the contravener, as specified in section 35 of the Act, can potentially provide any real economic remedy to the copyright owner for free and open source software. This is all the more true if the software was illegally included within proprietary software that was somewhat successful. However, the costs associated with such a litigation, especially lawyers' fees, are typically out of proportion to the amounts that can be recovered. Consequently, developers of free and open source software have a strong incentive to settle their disputes out of court.

[124] In addition to the civil remedies they may incur, violators of the *Copyright Act* are also committing a penal infraction.[116] This is perhaps the greatest protection granted to free and open source software, since anyone found guilty of copyright infringement is liable for a maximum fine of up to one million dollars and five years imprisonment.

7.2. Establishment of the chain of title

[125] Traditionally, software is the product of the efforts of a small group of programmers working together within a hierarchical organization. However, "The structure of work and communication in the hacker community is decentralized and distributed."[117] Indeed, the evolution of most free and open source software relies on voluntary contributions from its users. They become involved, from all over the world, as a function of their expertise and their specific needs.

---

[115] L. ROSEN, *op. cit.*, footnote 44, p. 274.

[116] *Copyright Act*, *op. cit.*, footnote 10, sect. 43.

[117] Eric S. RAYMOND, quoted in William C. TAYLOR, "Inspired by Work", (1999) 29 *Fastcompany* 200, source: <http://www.fastcompany.com/online/29/inspired.html>.

[126] The article "The Cathedral and the Bazaar"[118] by Eric Raymond, contains what has certainly become the best known description of this development model. The free software community is likened to a bazaar in which merchants interact in the open without being subordinated to an organizational structure. Conversely, traditional proprietary development is compared to a cathedral, where tasks are performed in a closed and hierarchical environment.

[127] However, the bazaar model does not precisely reflect the reality. All free and open source projects must have a minimum of organization.[119] Most frequently, this is provided by the project's promoter, who guides the evolution of the computer code, sets up procedures, and instils motivation in the developers. Furthermore, a structure is required so as to resolve conflicts that may occur when contradictory solutions are proposed. As a matter of fact, the bulk of the code for a given free and open source software project is often provided by a small set of promoters, the numerous occasional contributors providing only complementary assistance. Consequently: "Free Software development is less a bazaar of several developers involved in several projects [and] more a collation of projects developed single-mindedly by a large number of authors"[120].

[128] As a result, the principal contractual structure that this development model gives rise to takes the form of a chain of contracts. Under this linear structure, the licensor, the first licensee, and then numerous sub-licensees succeed each other.[121] In this configuration, rights are derived from the contracting party, who in turn guaranties having previously obtained a licence for the software. In reality, this is not the most prevalent configuration.

[129] In practice, the development of virtually all free and open source software can be best represented with a star, in which the project promoter is the core and the various contributors are rays.[122] From a legal perspective, it is first and foremost

[118]Eric S. RAYMOND, "The Cathedral and the Bazaar", (1998) 3 *First Monday*, source: <http://www.firstmonday.org/issues/issue3_3/raymond/index.html>.

[119]Charles CONNELL, "Open Source Projects Manage Themselves? Dream On", (2000) *Lotus Development Network*, source: <http://www-10.lotus.com/ldd/devbase.nsf/articles/doc2000091200>.

[120]Rishab GOSH et Vipul Ved PRAKASH, " The Orbiten Free Software Survey ", (2000) 5 *First Monday*, source : <http://www.firstmonday.dk/issues/issue5_7/ghosh/index.html>.

[121]C. ROJINSKY and V. GRYNBAUM, Les licences libres et le droit français, Propriété intellectuelles, July 2002/4, p.33.

[122]C. CARON, *loc. cit.*, footnote 22.

the link existing between each contributor and the promoter that affects the chain of title that must link the authors of the software to its users.

[130] First, some project promoters require from the contributors an assignment of all their rights over the code they submit. This is the case with the FSF in the framework of the GNU project.[123] In this situation, even when there are a large number of developers involved, the promoter is the sole owner of copyright over each contribution, as well as over the software as a whole. Thus, he is free to distribute it however he sees fit.

[131] Other promoters, on the other hand, settle for obtaining a grant through the use of a licence, which must be the same, or at least as permissive, as the one under which the software itself is distributed. This is the approach of the ASF, which requires of all contributors that they submit their code under a licence granting non-exclusive rights to perform all acts protected by copyright.[124] It follows that all contributors maintain the ownership of the rights over their contribution, and retain these rights even after is integrated into the software. The ASF, for its part, has all the necessary rights to re-license the software.

[132] Finally, most promoters take a considerably more informal approach. They do not impose any formality, and agree to incorporate any contribution submitted for that purpose into the software. A strict application of the *Copyright Act* would suggest that these operations are without any legal effect, the promoter not having acquired the authorization to exercise the rights attributed to contributors. However, it seems more appropriate to infer that an implicit licence was granted whenever this is possible. As to future licensees, each contributor retains some right over the software. It remains the case, however, that this form of contribution constitutes a weak link in the chain of title.

[133] The four situations described above do share one point. Once collected, the contributions are all combined together to form a standalone program. As of then, it becomes difficult to distinguish the part created by one contributor from those created by the others. The *Copyright Act* designates this type of creation as work of joint authorship.[125]

---

[123]FREE SOFTWARE FOUNDATION, "Information For Maintainers of GNU Software", (2005) *GNU,* source: <http://www.gnu.org/prep/maintain/maintain.html#Legal-Matters>.

[124]APACHE SOFTWARE FOUNDATION, "Individual Contributor License Agreement", 2005 *ASF*, source: <HTTP://WWW.APACHE.ORG/LICENSES/ICLA.TXT>.

[125]*Copyright Act*, *op. cit.*, footnote 10, sect. 2.

[134] Since the *Copyright Act* is mute on the effects of this statute, we must look to Quebec civil law. Under the QCC, most free and open source software is subject to undivided co-ownership, as defined in section 1010. The major consequence of this indivision is the existence of a single global copyright over the entire software. This also gives rise to an assumption of equality in the shares of the contributors and each one's authorization to perform acts affecting the software, provided these acts do not prejudice either its destination nor the rights of the other developers.[126] Thus, in the case of Tremblay v. Nguyen, the Superior Court of Québec confirmed the right of the owner of undivided rights to address the courts individually in the event of a violation of those rights.[127] Similarly, any licence granted by one of the undivided owners should be valid provided that the software was intended for distribution under a free and open source licence. Clearly, this indivision ceases to be effective when, as with the FSF, all shares are transferred into the hands of a single person.

[135] When this initial development stage is over, the chain of title is far from ended. Indeed, once software has been distributed on the Internet, free and open source licences favour the multiplication of successive grants over the software. At each stage, a fresh contributor (or group of contributors) agrees to grant rights over the newly added code, while retaining ownership over it. Thus, indivision continues to spread. This situation will not usually cause problems as long as the contributors continue to agree on the selected licence. However, there is a serious risk that a change of licence by one of the contributors may cause discontent amongst the others.[128]

[136] A major problem may arise if one of the links in this chain fails, i.e. if the grant of rights proves to be void. In that case, all subsequent licensees find themselves in an irregular situation, since any actions they may have undertaken with regard to the software infringed on the copyright owned by developers above the broken link.

[137] Two situations can give rise to this result. The first is nullity of one of the licences within the chain of title. The second is the illicit introduction into the software of code that is protected by the rights of a third party.

---

[126] L. CARRIÈRE, *loc. cit.*, footnote 18, p. 14.

[127] *Tremblay* v. *Nguyen,* (1997) 24 C.P.R. 289.

[128] L. ROSEN, *op. cit.*, footnote 44, p. 33.

[138] The invalidity of a free and open source licence may spring from several grounds. The clearest example may be a contribution submitted by an employee without the consent of his employer. Similarly, a Québec judge could refuse to recognize such a licence on the basis of the absence of a signature. In all events, the international character of the development of free and open source software has the effect that all links in the chain of title may prove to be invalid under the domestic law of a distinct country.

[139] The illicit introduction of code that is protected by the rights of a third party is precisely the basis of the current litigation opposing SCO and IBM. SCO alleges that IBM introduced copyrighted elements of UNIX into the Linux kernel. Aside from issues of breach of contract and divulging trade secrets that underlie the demand for compensation from IBM, it is precisely the claim that proprietary code has been incorporated into Linux that justifies SCO's position with respect to the thousands of users of this operating system[129]. Even if we may wonder whether, two years after the beginning of this case, SCO will be able to satisfy the burden of proof, their recourse is valid and this type of litigation is liable to recur in the future.

[140] Finally, a last element could render the chain of title of free and open source software even more complex. This is the case of distributions that gather various independently developed software packages into a single collection. This is the case, for example, with many Linux distributions, including the Red Hat version. The specificity arises from the fact that, in addition to the protection afforded individually to all included software, the distribution itself is protected. This copyright protection is recognized by both the *Copyright Act*[130] and jurisprudence,[131] in the context of protection provided to compilations. Free and open source software distributions are usually covered by a licence authorizing their use, modification, and redistribution. However, the exercise of these rights implies that special heed must be paid to compliance with both the licence of the compilation and those governing all of the software within it.

[141] Overall, the existence of a failure in the chain of title doubtlessly constitutes the greatest legal risk to users of free and open source software. Indeed, as soon as one of the links in the chain that should connect them to each of the developers is

[129] K. D. GOETTSCH, *loc. cit.*, footnote 66.

[130] Copyright Act, *op. cit.*, footnote 10, sect. 2.

[131] *Télé-direct (Publications) Inc.* v. *American Business Information, Inc.*, [1998] 2 C.F. 22, source: <http://www.canlii.org/ca/cas/fca/1997/1997fca10177.html>

missing, they are in a position of copyright infringement. Unfortunately, there is no way to easily establish the reliability of this chain of title, and users will generally only discover the offence when they are the target of legal proceedings. It is true that, even when there is a failure, the vast majority of copyright owners have no interest in raising that fact and consequently paralyzing the development of software which they, themselves, need. In this respect, the development model of free and open source software largely relies on the good faith of the developers involved.

[142] In light of the current practice, this risk must be put into perspective. After over fifteen years of widespread use of free and open source software, the lawsuit initiated by SCO is the only instance in which the chain of title of any such package was challenged. Furthermore, the same issue was always present with proprietary software, though on a significantly more limited scale given its relatively closed development model. Far from relying on new mechanisms, free and open source software only pushes the rules governing successive grants of intellectual property rights to their extreme limits.

7.3. Conflicts between licences

[143] In addition to risks related to failures in the chain of title, the method by which free and open source licences address intellectual property rights causes a certain number of difficulties when integrating code originating from different projects. On this matter, rather than reinventing the wheel each time a known problem recurs, the collective development model of free and open source software encourages developers to reuse existing code as much as possible. However, the numerous contradictory clauses found in various licences create serious limitations to such combinations, since licensees seeking to reuse are bound to comply with each one of them.

[144] Nonetheless, inasmuch as free and open source licences are concerned, no incompatibility limits the concurrent use of software. Thus, it always remains possible to install free and open source word-processing software (OpenOffice) over a proprietary operating system (Windows). Interactions between those tools do not infringe upon the terms of free and open source licences in any way—as they allow all kinds of uses without any particular restriction.

[145] Furthermore, even when software needs to be modified, this can often be done to meet the personal needs of the licensee. Indeed, most free and open source

licences only impose conditions on the redistribution of the software to third parties. Thus, an organization is authorized to integrate code licensed under the GPL into software whose code is not freely accessible, provided that the resulting software is confined to internal use. It is, however, generally considered better policy to avoid such situations, so as to head off risks potentially arising from an illicit distribution.

[146] Consequently, it is at the stage of distributing software created by combining source code released under different licences that incompatibilities between them are liable to become an issue. Then, the solution depends upon the type of licences involved.

[147] The most simple instance of this involves integrating free and open source software with software available in the public domain. Since this latter is subject to no conditions, it can be combined with any software, regardless of the licence under which it is distributed. The final product of this process is generally released under the free and open source licence that governs the other part of the initial code.

[148] The situation becomes a little more complicated when one of the programs involved is governed by a proprietary licence. In this case, the restrictive terms of the proprietary licence usually prohibit any modification to the code, and consequently any integration with free and open source software. Assuming, however, that it is allowed, such as when the integration is performed by the owner of the rights himself, free and open source licences impose conditions that vary depending on whether or not they are copylefted. First, since non-copylefted licences are extremely permissive, there is no obstacle to the integration of the code they govern into proprietary software, provided that their terms are respected. Adobe's Acrobat Reader software is an excellent example of this, its legal fact sheet reproducing a panoply of free and open source licences, as these latter require. Conversely, copylefted licences are always incompatible with proprietary licences, since the software resulting from this combination must necessarily be subject to supplementary constraints.

[149] Non-copylefted licences are usually not only compatible with each other, but also with all other types of licence.[132] In fact, the simplest ones (BSD, MIT) do not contain any clauses liable to create interactions with another licence. However, more elaborate non-copylefted licences may sometimes conflict with some copylefted licences. Thus, the FSF rejects contributions submitted under the Apache licence,

[132]A. ST. LAURENT, *op. cit.*, footnote **Erreur ! Signet non défini.**, p. 161.

owing to the additional terms imposed by this licence in comparison to the GPL, in particular with regard to suspension of software patents. Such a result is paradoxical, since the FSF invokes the incompatibility of the two licences while simultaneously agreeing with the concept underlying the offending clauses of the Apache licence.[133]

[150] As to copylefted licences, their specificity is in the stipulations that they apply to any software derived from code that was initially subject to them. This is spelled out in section 2 b) of the GPL:

> "You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License."

[151] Consequently, copylefted licences are completely incompatible with more restrictive licences, but also with all other copylefted licences. This is true even when two copylefted licences grant exactly the same rights, since each one requires that the resulting software must be released under its own terms. Consequently, copylefted code can only be integrated into code that is either in the public domain or released under a non-copylefted free and open source licence.

[152] However, some copylefted licences contain exceptions to that rule. This is the case, for example, with the MPL, which distinguishes between the initial code and subsequent contributions, and allows another licence to apply to modifications to the contributions. In this context, it proves possible to comply with the terms of another copylefted free and open source licence. However, the code subject to that exception must be identified with a great deal of care.

[153] The following table illustrates the interactions described above:

[133] FREE SOFTWARE FOUNDATION, "Various Licenses and Comments about Them", (2005) *FSF*, source: <http://www.fsf.org/licensing/licenses/license-list.html#GPLIncompatibleLicenses>

	Public domain	BSD	Apache	GPL	MPL	Proprietary
Public domain	yes	yes	yes	yes	yes	yes
BSD	yes	yes	yes	yes	no	no
Apache	yes	yes	yes	no	no	no
GPL	yes	yes	no	yes	no	no
MPL	yes	no	no	no	yes	no
Proprietary	yes	no	no	no	no	no

Table 1: Compatibility of the most common licences

[154] Unfortunately, the information included in this table cannot simply be applied mechanically. Indeed, the licences listed are often used as models for the elaboration of new ones, the terms of which can differ to a greater or lesser extent from those of the original. This course of action is generally not recommended, since each new licence generates its own set of incompatibilities with its precedents.[134]

[155] Owing to constraints created by these conflicts between licences, software is now often offered under several different licences.[135] Under these conditions, it becomes possible to choose the licence that is best suited to the intended use for the software. This duality of licences is made possible by the very nature of software licences, which only concede a non-exclusive right to perform certain acts that would be prohibited otherwise. Thus, nothing prevents the licensor from authorizing licensees to perform the same acts under different terms. This also means he can always provide permission to waive the terms of a free and open source licence. This is provided for in section 10 of the GPL:

> "If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally."

[156] These special requests are often granted, since the developers of free and open source software are very interested in encouraging use of their software.

[134]David A. WHEELER, "Make Your Open Source Software GPL-Compatible. Or Else.", (2005) *D. Wheeler*, source: <http://www.dwheeler.com/essays/gpl-compatible.html>.

[135]Julien LINSOLAS, "Le statut juridique du logiciel libre", (2003) 2 *Droit NTIC* 2, source: <http://www.droit-ntic.com/pdf/revuefevrier2003.pdf>

Moreover, since the goal of these licences is to facilitate the free dissemination of the software code, it is paradoxical that their profusion may hamper this sharing.

[157] Also, as in the case of identifying the chain of title, compilations of free and open source software pose further difficulties in the matter of evaluating the compatibility of the terms of their licences. First, in light of the structure of free and open source licences, software of this type assembled within a single compilation need not feature fully compatible licences. Indeed, inasmuch as the various software programs in the compilation remain independent, i.e. are not all integrated together, their distribution simply requires that each one be accompanied by its own licence. However, in addition to these licences, there is a licence governing the compilation as well. In consequence, any integration of multiple compilations is vulnerable to all the conflicts exposed above. This explains why a compilation subject to a BSD licence could be combined with an Apache compilation, while a GPL compilation could not—without regard for the licences governing their components. In addition, it always remains possible to take one or several software elements out of a compilation, or even to integrate them with those from other compilations. In this case, the originality of this new compilation makes compliance with the licences of the preceding compilations superfluous.[136]

[158] Finally, software libraries are also subject to some particularities when conflicts between licences are invoked. This is attributable to the fact that a library might be subject to a different licence from the original software. To the extent that one these two elements is copylefted, it is likely that the software resulting from their interaction must be subjected to this licence. However, this result is not definitive, since it depends on the interpretation of each licence and the answer is far from being straightforward.[137] Even the LGPL, designed specifically to resolve this difficulty, raises numerous questions. Thus, until the point is clarified, the best solution would certainly be to limit the use of libraries to those whose licences are clearly compatible with that of the original software.

7.4. Software patents

[159] So far, we have only examined the protection of free and open source software from the perspective of copyright. However, intellectual property includes

[136] L. ROSEN, *op. cit.*, footnote 44, p. 242.

[137] L. ROSEN, *op. cit.*, footnote 44, p. 124.

another legal framework that could apply to software: patent law. The fundamental purpose of patents is to protect inventions that are useful, original, and not obvious. Just as in the case of copyright, the point is not to protect the right over an idea, but rather to protect its physical implementation, i.e. the manufacture of an object that materializes it. Thus, the invention is patentable while the discovery is not.

[160] The federal *Patent Act* defines an invention as follows:

> "...any new and useful art, process, machine, manufacture or composition of matter, or any new and useful improvement in any art, process, machine, manufacture or composition of matter."[138]

[161] Even though software usually meets these criteria, section 27(8) of the same Act provides for an exception when the invention is based on simple scientific principles or abstract theorems. Thus, we need to turn to the Manual of Patent Office Practice, of which a new edition was published in February 2005, for the interpretation given to this provision by the federal authorities responsible for granting patents in Canada.

> "Software expressed as lines of code or listings is considered to be a literary work under the *Copyright Act*. Software in the form of a data model or an algorithm is automatically excluded from patentability under subsection 27(8) of the *Patent Act*, in the same manner as a mathematical formula, and is considered to be equivalent to a mere scientific principle or abstract theorem. However, computer related subject matter is not excluded from patentability if the traditional criteria for patentability are satisfied. Software that has been integrated with statutory subject matter may be patentable."[139]

[162] Thus, software is generally excluded from the field of patentable inventions in Canada. This position is largely the result of the decision of the Federal Court of appeal of Canada in the Schlumberger Canada Ltd. v. Commissioner of Patents case.[140] On this occasion, the court ruled that there was nothing innovative in using a computer to perform calculations, and that a mathematical formula was nothing more than a simple scientific principle or an abstract theorem. It added that these same calculations, had they been performed by a human being, would have constituted a sequence of mental or intellectual operations, which are not patentable in Canada.

---

[138] *Patent Act*, R.S. 1985, c. P-4, sect. 2, source: <http://www.canlii.org/ca/sta/p-4/sec2.html>.

[139] Canadian Intellectual Property Office, "Manual of Patent Office Practice" (2005) CIPO, source: http://strategis.ic.gc.ca/sc_mrksv/cipo/patents/mopop/chap16-e.html.

[140] *Schlumberger Canada Ltd.* v. *Commissioner of Patents,* (1981) 56 C.P.R. 92nd) 204 (F.C.C.).

[163] It remains, nonetheless, that since this landmark decision many requests for patents covering software have been accepted. In most of these cases, the patents were granted on software that was integrated into customarily patentable elements, such as printers,[141] telecommunications systems, [142] or elevators.[143] In some cases, "pure" software was patented on the sole argument that it could be linked to at least one physical component, as in the Motorola case.[144] The jurisprudence thus suggests a very liberal trend in the decisions of the Patent Appeal Board in the matter of software patents. It is now recognized that it suffices to translate software into a device, by linking it to some physical element, to make it patentable.[145]

[164] However, the principle of non-patentability of software remains the dominant position in the vast majority of countries around the world. In fact, only the United States and Japan officially recognize software patents.[146] In Europe, the European Patent Office accepts them despite the exception expressly provided for by the *European Patent Convention*, which specifies that inventions exclude "schemes, rules and methods for performing mental acts, playing games or doing business, and programs for computers."[147] Europeans have been debating the relevance of this provision for over three years in the context of various directives proposing the regulation of software patents.

[165] This openness of administrative agencies to software patents, which we observe in Europe as well as in Canada, can largely be traced to pressure exercised by the U.S. government on the international scene.[148] Indeed, the United States actively promotes software patents so as to better ensure protection for the intellectual property rights of its main software publishers.

---

[141] *Application for Patent of International Business Machines Corp.*, (1984) 6 C.P.R. (3d) 99.

[142] *Re Application for Patent of Janssens*, (1984) 6 C.P.R. (3d) 213.

[143] *Re: Application for Patent of Westinghouse Electric Corp.*, (1985) 6 C.P.R. (3d) 58.

[144] *Re Application for Patent of Motorola Inc.*, (1999) 86 CPR (3d) 71 and 76.

[145] Louis-Pierre GRAVELLE, "De la vie aux ordinateurs – développements récents en matière de brevetabilité des organismes vivants, des méthodes de traitement et des technologies informatiques", (2000) 138 *Développements récents en propriété intellectuelle* 104, p. 106, source: <http://www.robic.ca/publications/Pdf/254-LPG.pdf>.

[146] Russell MCORMOND, "A Review of Software Patent Issues", (2003) *Digital Copyrigth Canada*, source: <http://www.flora.ca/patent2003/>.

[147] *Convention on the Grant of European Patents*, 1973, art. 52, source: <http://www.european-patent-office.org/legal/epc/e/ar52.html#A52>.

[148] Brian KAHIN, "Information Process Patents in the US and Europe: Policy Avoidance and Policy Divergence", (2003) 8 *First Monday* 3, source: <http://www.firstmonday.dk/issues/issue8_3/kahin/#author>.

[166] However, in addition to the problems they create for the entire software industry, patents constitute a significant danger to the future of free and open source software. Several factors underlie this risk:

- Patents allow the protection of techniques that are trivial, but are sometimes necessary to the developers of free and open source software;
- The possibilities for rewriting an algorithm in an unprotected form diminish in proportion to the proliferation of software patents;
- As the number of patents increases, it becomes nearly impossible to avoid using one of them without knowing, as is currently occurring in the United States;
- The costs of registering patents, as well as the lawyers' fees required to enforce them, benefit large publishers with an anti-competitive arsenal over the developers of free and open source software—mostly small- and medium-sized companies and individuals—which would never be able to afford such means;
- Patentability of file formats and communications protocols makes it possible to cash in on interoperability, which is generally unattainable for developers of free and open source software;
- Patent infringement is easier to prove against the developers of free and open source software, since the code of this software is freely available for examination.

[167] Consequently, growth and technological development through free and open source software requires maintaining the principle of non-patentability of software.[149] Within Québec, this amounts to ensuring the effective application of the conclusions of the Schlumberger decision. This solution appears all the more attractive since even some opponents of free and open source software recognize that patents do not represent an effective means for protecting software.[150]

---

[149]PRICE WATERHOUSE COOPERS, Rethinking the European ICT Agenda: Ten ICT-Breakthroughs for Reaching Lisbon Goals, (2004) *Price Waterhouse Coopers*, p. 52, source: <http://www.pwc.com.nyud.net:8090/Extweb/pwcpublications.nsf/docid/EC6DE73A846581CE80256EFD002E41FB/$file/pwc_rethinking_european_ict_agenda.pdf>.

[150]Mathias STRASSER, "A New Paradigm in Intellectual Property Law? The Case Against Open Sources", (2001) *Stan. Tech. L. Rev.* 4, par. 45, source: <http://stlr.stanford.edu/STLR/Articles/01_STLR_4/article.htm>.

8. Contract

[168] When a licensee accepts the terms of a free and open source licence, a contractual relationship is established with the owner(s) of the software's copyright. This contract, of variable legal designation, primarily serves the interests of the licensor, and is designed to be replaced by the stricter rules of intellectual property should one of its clauses be violated. Conversely, the licence is the principal resort of the licensee in the event of faulty software or default on behalf of the supplier. This is why examination of the contractual context essentially focusses on establishing criteria for identifying the liabilities of free and open source licensors. This, in turn, requires demonstrating that the licensor neglected to perform an obligation placed on him by the licence contract or by the law. Because free and open source licences are specifically designed to limit the licensor's civil liability, the burden of proof may prove difficult to satisfy. The consumer protection regime may sometimes facilitate establishing this liability.

8.1. The legal designation of the licence contract

[169] A critical feature of free and open source licences is that they are dictated by the licensors. Most frequently, they integrally reproduce a standard contract (such as the GPL) and impose it on everyone wishing to use their software. Consequently, *a priori* the licensee has no opportunity to negotiate its terms. Those licences can thus be construed as contracts of adhesion.[151] This designation does not, in and of itself, affect the rights and obligations of the parties. However, it may affect the validity of exemption clauses in the contract. They may, in fact, be deemed abusive clauses, which are proscribed under section 1437 of the QCC.

[170] On the other hand, the issue of the legal designation of free and open source licences, i.e. the identification of an appropriate nominate contracts, is of great importance. In fact, it conditions the application of legal rules relative to the envisioned operation and can thus impact the rights and obligations of the parties. Unfortunately, even in the case of proprietary software, the issue of the legal designation of software licences has not found a conclusive and unanimous solution.

[171] Part of the legal litterature favours the legal designation of sales contracts for software that is distributed with hardware, since an accessory follows its

[151] M. CLÉMENT-FONTAINE, *loc cit.*, footnote 51, sect. 26.

principal.[152] This was the gist of the 1985 ruling by the Québec superior court in the Olier, Grisé & Cie Ltée v. Équipements de bureau Maskoutan Inc. case,[153] which designated a software licence as a sales contract pure and simple, since the software was an integral part of the supplied hardware. Similarly, several judgements by the Court of Appeals in Paris[154] applied the sales regime to a set of contract, reasoning from the dependence of the contract under which the software is made available (the licence) on the contract for the provision of hardware that supports it. This case addresses a latent defects action with respect to the warranty provided with the good.[155]

[172] Similarly, in the matter of packaged software, the legal framework governing sale is preferred by some authors.[156] They adopt a "materialist" approach, viewing the licence regime from the perspective of the sale of individual copies of the software. Consequently, in a general sense the sales regime could be applied to contracts (or groups of contracts) that provide for supplying computer systems including hardware, packaged software, even specific development, on condition that the hardware and packaged software constitute the bulk of the supplied system. However, this designation of sale is not satisfactory from the perspective of intellectual property, since most packaged software licensors retain their rights so that the contracts do not transfer any immaterial rights.

[173] Owing to this absence of transfer of any right over the software, virtually all authors consider the licence contract as similar to the lease contract.[157] This designation appears appropriate in regard to the "economics" of the licence contract: The licensor grants the licensee the enjoyment of an intangible thing (the software) in exchange for the payment of a rent (fee) over a limited period of time.[158] Furthermore, the notion of a lease allows the licensor to limit the acts that the licensee is authorized to perform, so that the latter cannot freely dispose of the software. It also allows the licensor to receive regular compensation over the

[152]Philippe LE TOURNEAU, "Très brèves observations sur la nature des contrats relatifs aux logiciels", (1982) 1 *JCP* 3078.

[153]*Olier, Grisé & Cie Ltée* v. *Équipements de bureau Maskoutan Inc.*, [1985] C.S. 680-683.

[154]CA Paris Oct. 3, 1989, Sté SNDA c/ Sté ICL France, Cahiers Lamy, Feb. 1990(K); CA Paris Feb. 8, 1990, Sté ICL France c/ Sté Lanvaux-Ronsard, juris-data No. 20232; CA Paris, 25th ch. B, June 22, 2001; see also CA Bastia, ch. civ., Nov. 19, 2002, Juris-Data No. 2002-00772.

[155]*Québec Civil Code*, sect. 1641.

[156]Ean MACKAAY, "Le marché du progiciel – licence ou vente?", (1994) 6 *Cah. Prop. Int.* 401.

[157]Michel VIVANT et al., *Lamy droit de l'informatique et des réseaux*, Lamy, Paris, 2001, no. 840, p. 522.

[158]*Québec Civil Code*, *op. cit.*, footnote 11, sect. 1851.

duration of the software usage, in the form of a licence fee. This designation thus appears to be the most appropriate for the licence contract, even though the litterature does raise several reservations. Indeed, the analogy falls short in that the licensor, unlike the leasor, does not suffer the dispossession of the goods that he makes available, especially since he reserves the right to "lease" it to other users.

[174] This is surely the reason why Québec courts generally prefer to treat software licences as neither in the nature of a sale, nor in the nature of a lease, but rather as a innominate contract conferring an individual right of use. This solution clearly follows the Informatique L.G.A. Inc. v. Compagnie d'arrimage de Québec Ltée case,[159] in which the judge emphasized that the licence donot transfer any real right, but simply an individual right to use an intangible thing. This solution has a consequence that is very important in practice: Since neither the sales, nor the lease, regime apply to the licence, the legal warranties that they provide for are automatically inapplicable.

[175] However, this jurisprudence has been qualified since 1994 by the decision in the Unicel Inc. v. Contalitec Informatique Inc. case.[160] On this occasion, the Court of Québec confirmed the analysis based on the *sui generis* licence contract, but added that it features certain characteristics of the lease (i.e. rental) contract. The judges deemed that the licensor must be subject to the obligations imposed on the leasor by the QCC, i.e. the legal warranties associated with the lease agreement.

[176] In the matter of the development and provision of specific software, Québec jurisprudence retains the designation of an innominate contract,[161] in contrast with French law, which treats it as a contract for services. Indeed, under French legal litterature it is necessary to distinguish the case of specific software, which may be free and open source software specifically commissioned by the licensee.

[177] Furthermore, when the economic aspect of the contract are dual (the supply of hardware and the adaptation of software), Québec courts have, on occasion, even gone to the length of segregating the designation of the operation into two parts, with a sales contract for the hardware, on the one hand, and an innominate contract for the specific software, on the other.[162]

---

[159] *Informatique L.G.A. Inc.* v. *Compagnie d'arrimage de Québec Ltée*, [1991] R.J.Q. 1767 (C.Q.).

[160] *Unicel Inc.* v. *Contalitec Informatique Inc.*, (1994) J.E. 94-1910 (C.Q.).

[161] *Sillons Le Disquaire Inc.* v. *Datagil Informatique Inc.*, (1998) J.E. 98-1148 (C.Q.).

[162] *Sillons Le Disquaire Inc.* c. *Datagil Informatique Inc.,* op cit., footnote 161.

[178] Finally, the legal designation differs again when the software is distributed free of charge. In this case, since the licensor receives no compensation, application of sales or lease regimes is precluded. Under Québec civil law, it is possible to consider this contract type as a gratuitous loan. The obligation for restitution underlying the contract thus becomes irrelevant, since software is an immaterial good that can easily be replicated.[163]

[179] On the whole, this heterogeneous and variable body of opinions in the litterature reveals to what extent the software licence contract is difficult to identify with one of the nominate contracts provided for by the QCC. However, this operation of legal designation is far from trivial. in such a case, contractual freedom is, in fact, doubly constrained: On one hand, the shared intention of the parties may not impair the public order rules specified by a given contract regime and, on the other hand, it is frequently the case that, in order to identify this shared intention, the designation of the contract allows the judge to enforce the applicable rules in the event of ambiguity, omission, or "grey zones" in the licence.

[180] Finally, designation as innominate contracts appears quite appropriate to software licences. This is justified both by their specific object (immaterial good) and by the content of the rights conferred by the contract (the right to limited and non-exclusive use).[164] However, this solution is not definitively set in the jurisprudence, and some courts still prefer to apply the sales or lease regimes to licences in order to introduce the waranties implied by these regimes.

[181] Applied to free and open source software, the legal designation of the licence contract thus remains variable, depending on the specifics of each case. However, this designation is liable to have a determinant impact on the rights and obligations of the parties, as well as setting the regime of warranties legally applicable. Thus the licensee must be particularly attentive to the designation that may apply to the licences he is agreeing to, so as to anticipate the legal risks associated with the designation in each case.

8.2. The obligations of the parties

[182] The object of software licences is to define the terms and limitations under which the licensee may use the software in question. This is why they generally

[163]M. CLÉMENT-FONTAINE, *loc cit.*, footnote 51, sect. 24.

[164]Frédérique TOUBOL, *Le logiciel : Analyse juridique*, Feduci – L.G.D.J., Paris, 1986, p. 128.

contain more obligations expressly imposed on the licensee than on the licensor, especially they can be considered as contracts of adhesion. Free and open source software licences are no exception to this rule. They even precisely delineate the conditions under which the licensee may exercise the rights bestowed on him, while totally omitting any obligations of the licensor. Consequently, when the licensee is in a position to privately contract a free and open source licence with a supplier, it is in his interest to balance the contract, by increasing the obligations that the licence expressly puts on the copyright owner. However, if no negotiation is possible as it is more commonly the case, it remains that the licensor must comply with certain legal requirements.

The specific obligations of the licensee

[183] These obligations vary with the free and open source licence under consideration. In general, they assume the form of "rights with responsibilities," i.e. the authorization to perform a specific act while satisfying certain conditions or obligations. Thus, for example, section 2 of the GPL stipulates:

> « You may modify your copy or copies of the Program or any portion of it (…) provided that you also meet all of these conditions: a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change. b) You must cause any work that you distribute or publish (…) to be licensed as a whole at no charge to all third parties under the terms of this License. c) If the modified program normally reads commands interactively when run, you must cause it, (…) to print or display an announcement (…) ».

[184] Thus, obligations place on the licensee are most commonly "obligations to do" (positive duties), but contracts may also include express "obligations not to do" (negative duties). For example, section 4 of the GPL states:

> « you may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. (…) ».

[185] Finally, to these specific obligations must be added the obligation to pay the fee if the licence is granted under an onerous contract.

[186] Compliance with these obligations is primarily assessed as a question of fact, with the outcome depending on the actions of the licensee. Furthermore, the only provision for sanction that generally applies in the event of non-compliance with the terms of the licence is the revocation of the grant previously established. Under these circumstances, Québec civil law is superseded by intellectual property law.

The legal obligations of the licensor

[187] The law and jurisprudence charge the licensor with the obligation to deliver the free and open source software, as well as with the associated disclosure requirements and obligation of warranty.

[188] As the litterature specifies, the essential obligation of a supplier is to deliver the object of the contract[165]. In the case of free and open source licences, this means that the licensor should make the software available to the licensee. This delivery may assume several forms: delivery on a physical medium or online delivery through download.

[189] The principal issue associated with the obligation to deliver relates to making available the source code of the free and open source software. In principle, a licensee who accepts the terms of a free and open source licence has a legitimate expectation of receiving the code. Furthermore, copylefted licences obligate the licensee to provide the source code himself if he ever redistributes the software. However, this obligation is due to the person who provided him the software, and not to his own licensees.[166] Owing to the very nature of free and open source licences, we may consider that providing the source code is one of the essential obligations placed on the licensor—so that failure to comply with this obligation would render the contractual framework meaningless. A licensee who does not receive this code can invoke the contractual liability of the licensor, since the code is vital for the exercise of rights that are conferred on him by the licence. Conversely, the binary code is less relevant. It is not required because it does not make modifying the software possible, and it can always be recreated by compiling the source code. All things considered, it may be thought that the objective of free and open source licences is precisely the delivery of the software in the form of source code. From this perspective, the obligation to deliver remains unfulfilled as long as it is not made available.

[190] The obligation to deliver may also be interpreted to include the software documentation as an accessory. Among others, this position has been taken by French courts in dealing with proprietary software.[167] It is far from certain that this

---

[165]M. VIVANT et al., *op. cit.*, footnote 157, no. 1282, p. 734.
[166]FREE SOFTWARE FOUNDATION, *loc. cit.,* footnote 7, preamble, source: <http://www.gnu.org/licenses/gpl.html>.

[167]Philippe LE TOURNEAU, *Théorie et pratique des contrats informatiques*, Éditions Dalloz, Paris, 2000, p. 97.

principle can be applied to free and open source licences, since it violates the very spirit of these agreements. Indeed, the documentation and the computer program are considered as independent elements in these contracts. While the licensee of proprietary software expects the documentation to follows the software, a free and open source software licensee would be more likely to look for documentation within the community of developers using the software. Since the scope of the obligation to deliver is subordinated to the willingness of the parties, documentation should thus be excluded in matters of free and open source software. Furthermore, the obligation to deliver must be interpreted in light of the *Copyright Act*. This act defines computer programs in terms of a series of instructions designed to be used by a computer.[168] Documentation, didactical material, and flowcharts do not readily fall under this definition. These elements may nonetheless benefit from their own independant copyrigth protection, and should in this case become the object of an independent contract. Moreover, while some authors define the notion of source code so as to include the comments that are necessary to understand it,[169] free and open source licences exclusively refer to the computer program. Thus, it should not be possible to hold a licensor liable for any absence of comments within the source code.

[191] The second essential obligation placed on the licensor deals with the information he must provide to the licensee. Indeed, civil law sometimes places a basic disclosure obligation on the shoulders of the contractor possessing certain information that may be useful to the other party.[170] In the framework of computer contracts, the litterature and the jurisprudence have generally imposed a fairly broad obligation to disclose, inform, advise on the service supplier. In the field of free and open source software, the information that the licensor must provide to the licensee generally pertains to the terms and risks[171] associated with using the software (system requirements for its proper functioning and the main bugs). Similarly, as long as the first official version of the software is not released, it appear important for the licensor to clearly inform the licensee that he use a preliminary version (beta version). In the event of non-compliance with this obligation to inform,

[168]*Copyright Act, op. cit.*, footnote 10, sect. 2.

[169]Hervé CROZE, Franck SAUNIER, *Logiciels : retour aux sources*, JCP ed. Générale, Paris, 1996, doctrine 3909, s. 7, p. 94.

[170]Claude LUCAS DE LEYSSAC, "L'obligation de renseignements dans les contrats", *coll. l'information en droit privé*, LGDJ, Paris, 1978, p. 305.

[171]M. VIVANT et al., *op. cit.*, footnote 157, no. 924, p. 556.

the licensor may be held liable to repair the damages he has caused, especially if the free and open source licence was concluded as a onerous contract.

[192] The licensor may also be bound to warrant the software he distributes. These warranties may be express, though this is very rare in the case of free and open source licences. Nonetheless, he may remain liable because of implicit warranty obligations specified in the law.

[193] When developers promote the features and functionalities of their software, it is possible that courts will consider those declarations as obligations of warranty (explicit warranties). In these circumstances, the licensor seeking to avoid civil liability must be prudent in how he presents the free and open source software to the licensee, since any statements made can be construed as commitments. For example, this could be the case if he claims that the software is compatible with some recognized standard. To benefit from these warranties, the licensee must prove that the relevant declarations fell under the ambit of the contract, and consequently have binding force upon the contracting party. In light of circumstances, such burden of proof may be difficult to establish. Nonetheless, legislation sometimes alleviate the burden on the licensee by expanding the scope of the agreement existing between the parties. Québec's *Consumer Protection Act* contains such provisions.[172]

[194] Licensors' contractual obligations are not the only elements the licensee may invoke in support of the litigation he may initiate in the event of damages. In many cases, he may also benefit from one or several implicit legal warranties. These are automatically inserted by the law into the contractual sphere in order to re-establish balance among the obligations of the parties.

[195] These warranties are specific to each legal regime, and thus vary with the designation given to the contract. Among the many implicit warranties in existence, three are likely to be applied to free and open source licence contracts.

[196] The warranty against eviction is the first. It functions as a general principle and finds application in the context of sales[173] and lease[174] contracts. Concretely, it assures the licensee that his use of the software will not be disrupted, either in fact or in law, by the contracting party or any third party. This guarantees that the

[172] *Québec Consumer Protection Act*, op cit., footnote 79, sect. 41.

[173] *Québec Civil Code*, *op. cit.*, footnote 11, sect. 1723 and f.

[174] *Québec Civil Code*, *op. cit.*, footnote 11, sect. 1858.

licensor has not confered to the licensee a right that has already been assigned. Thus, it protects the licensee against any attempt to default on the object of the licence or to deprive him from using the software, provided he has not overstepped the rights conferred upon him.[175] This also ensures that the licensor is liable for any copyright infringement litigation initiated against the licensee by a third party.

[197] By its very nature, this warranty is particularly effective for protecting the licensee. Its most common application in the matter of software deals with interferences from third parties. There is, in fact, a risk that the use of free and open source software may violate the intellectual property rights of third parties, even unbeknownst to its developers. Nonetheless, the consequences arising from this type of situation are mitigated by the legal warranty against eviction, which allows the licensee—to the extent that this interference with his enjoyment of the free and open source software causes him damages—to claim damages or to call the licensor in warranty.[176]

[198] In any event, should the enjoyment of the licensee be restricted, French legal litterature deems that "the problem could not occur" inasmuch as:

> "the legitimate user of the software obtains from the law itself the right to perform acts that would, in the normal course of events, fall under private right. A client having obtained access to software from a contracting party who he believes owns the rights must, it appears, be considered a legitimate user. This should suffice to shelter him from any claims aimed at depriving him of the right to use the software."[177] (authors' translation.)

[199] Thus, the licensee who is deemed a legitimate user appears to be relatively well protected by the warranty against eviction, for as long as the licensor is solvent and compellable.

[200] Québec law provides for a second implied warranty mechanism: the warranty against latent defects. This essentially applies to sales[178] and lease[179] contracts. It may also apply in the context of loan agreements, but only if the lender was aware of the defect in the lent item. This warranty protects the licensee when a defect affects the software and renders it unsuitable to the use for which it was

[175]M. CLÉMENT-FONTAINE, *loc cit.*, footnote 51, sect. 24.

[176]Stephen M. MCJOHN, "The Paradoxes of Free Software", (2000) 9 *Geo. Mason L. Rev.* 25, 35.

[177]A. LUCAS, J. DEVÈZE, J. FRAYSSINET, "Droit de l'informatique et de l'internet," PUF, *coll. Thémis Droit privé*, No. 760.

[178]*Québec Civil Code*, *op. cit.*, footnote 11, sect. 1726 and f.

[179]*Québec Civil Code*, *op. cit.*, footnote 11, sect. 1854.

intended. Several conditions govern the application of this warranty. First, the defect must be intrinsic to the thing that is the object of the contract and it must predate its conclusion.[180] The defect must be latent, i.e. it must not be apparent at the time of delivery and it must not be known to the licensee at the time at which he accepts the terms of the licence. With regard to this conclusion, the QCC defines the word "apparent" as "a defect that can be perceived by a prudent and diligent buyer without any need of expert assistance."[181] Thus, the evaluation of software once it is delivered to the licensee must have been reasonable under the circumstances,[182] which judges would evaluate on the basis of factors such as the technical knowledge of the licensee, the nature of the software, and the claims, possibly misleading, which may have been made prior to the sale. Moreover, the defects must render the item unfit for the use for which it was intended or so diminish its usefulness that the buyer would not have bought it or paid so high a price if he had been aware of them. In terms of software, this type of defect corresponds to a major flaw. It may be, for example, a bug affecting one of the essential functionalities of the software that keeps it from working properly. Evaluation of the defect occurs *in abstracto*, i.e. according to an objective test that accounts for the normal functioning of software, the inconvenience caused to the licensee by the defect, and the potential repair costs incurred.[183] The defect in question may thus be purely functional, so free and open source software that is simply malfunctioning could be deemed to possess a latent defect according to the wording of the QCC.[184]

[201] In the event that application of this warranty is accepted, all licensors are bound by it, unlike in most common law jurisdictions[185] where it only covers merchant sellers. That being said, when the sales regime is applicable, the defect will be assumed to have existed at the time at which the contract was concluded if the licensor is, in fact, a professional seller.[186] Moreover, it is worth noting that the

---

[180]Denys-Claude LAMONTAGNE, *Droit de la vente*, Éditions Yvon Blais, Cowansville, 1995, p. 97.

[181]*Québec Civil Code*, *op. cit.*, footnote 11, sect. 1726. 2.

[182]Jurisprudence appears to require that the examination was serious and careful, though it may also have been conducted rapidly and not in depth, cf. *Trottier* v. *Robitaille*, (1994) J.E. 94-1591 (C.A.).

[183]*Averback* v. *Meunier*, (1992) J.E. 92-941 (C.S.); Pominville v. Demers, [1990] R.D.I. 97 (C.Q.); Eldon Industries Inc. v. Eddy Metal Products Co., (1990) J.E. 90-822 (C.A.).

[184]The defect need not be a breakdown of material, cf. *Bosa-Chatigny* v. *Roberge*, [1990] R.L. 1 (C.A.).

[185]See, for example, the *Sale of Goods Act*, R.S.O. 1990, c. S.1, sect. 15, which uses the concept of "merchant sellers."

[186]Québec *Civil Code*, *op. cit.*, footnote 11, sect. 1729.

warranty against latent defects applies equally to the developer, the distributor, the supplier, or the importer of the free and open source software.[187]

[202] The third, and last, implicit warranty that may find application to the provision of free and open source software is the warranty of fitness. Under civil law, this warranty has long been considered an component of the obligation to deliver[188], though this is nowhere stated explicitly. The warranty of fitness obligates the licensor to provide free and open source software that meets the specifications, needs, and specific objectives of the licensee, provided that these elements were inserted in the contractual sphere. In light of this last condition, recourse to the warranty of fitness finds limited applicability in the case of free and open source software, as long as the licence features the characteristics of a contract of adhesion. Conversely, it may be applicable if the licence is negotiated, or if free and open source software is provided under a broader contractual framework within which the licensee did express specifications, or if the free and open source software was specifically commissioned.

[203] To the extent that implicit legal warranties may be an integral part of free and open source licence contracts, subject to the legal designation potentially imputed to them by the courts, Québec law require the developers to design software possessing certain basic qualities. It is precisely for purposes of aliviating these warranties that exemption clauses are systematically included in free and open source licence contracts.

8.3. Exemption clauses

[204] Implicit warranties constitute the principal risk to free and open source software licensors in terms of contractual liability. Consequently, they seek to avoid their application by extensively using contractual disclaimers. This is the main feature shared by all free and open source licences.[189]

[205] Thus, for example, the BSD licence contains the following clause:

> "This software is provided by the author ''as is'' and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no

[187] *Québec Civil Code*, *op. cit.*, footnote 11, sect. 1730.

[188] F. TOUBOL, *op. cit.*, footnote 164, p. 131.

[189] Bruce PERENS, "The Open Source Definition", in DIBONA, OCKMAN and STONE (dir.), *Open Sources: Voices from the Open Source Revolution*, O'Reilly, 1999, source : <http://www.oreilly.com/catalog/opensources/book/raymond.html>.

> event shall the author be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage."[190]

[206] Similarly, section 11 of the GPL specifies:

> "No warranty" "Because the program is licensed free of charge, there is no warranty for the program, to the extent permitted by applicable law. Except when otherwise stated in writing the copyright holders and/or the other parties provide the program "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the program is with you. Should the program prove defective, you assume the cost of all necessary servicing, repair or correction."

[207] However, Québec law does not always allow contracting parties to entirely exclude civil liability. Moreover, even when exemption clauses are allowed, the courts generally reserve the right to limit their scope in light of the circumstances. Consequently, disclaimers found in free and open source licences are not always enforceable and may sometimes be without any binding effects.

[208] In Québec civil law, the validity of exemption clauses or disclaimers depends on the designation imputed to the contract. The sales regime is the only one, under certain circumstances, to proscribe clauses that void warranty, in particular when the seller is a professional.[191] Conversely, they are generally allowed in the case of leases. As to warranties against eviction, they make it impossible for the licensee to claim damages and interest, but his right to resiliate the licence or to lower the price is unaffected.[192] In the matter of latent defects, nothing stops the parties from placing the full brunt of the risks on the shoulders of the licensee. This last principle also applies to loans, to the extent that the warranty against latent defects finds application.

[209] However, exemption clauses will never have any effect if the licensor committed a gross or delibarate fault.[193] Good faith and the absence of any intention

---

[190]OPEN SOURCE INITIATIVE, *loc. cit.*, footnote 30.

[191]E. MACKAAY, *loc cit.*, footnote 156, 415.

[192]Marcel PLANIOL, Georges RIPERT, *Traité élémentaire de droit civil*, v. 2, 10th ed., L.G.D.J., Paris, 1926, no 525.

[193]Jean-louis BAUDOUIN, Pierre-Gabriel JOBIN, *Les obligations*, 5th Ed., Éditions Yvon Blais, Cowansville, 1998, no 872, p. 707.

to cause harm on the part of the licensor is thus a determinant element in the assessment of the validity of liability disclaimers.

[210] In Québec, exemption can also be ineffective in relation to certain types of damages. Thus, to "exclude or limit his liability for bodily or moral injury caused to another" is prohibited by the QCC.[194] Disclaimers may nonetheless remain useful under these circumstances, since they are equivalent to a warning of danger, which may eventually reduce the liability of the licensor.

[211] It is also possible that a given clause is declared void in relation to a specific situation. Indeed, courts tend to interpret them restrictively, and generally reserve the right to mitigate their effects.[195] Several elements could be taken into consideration when such a decision is made:

- The parties' bargaining power;
- The available options;
- The benefits conferred in exchange for the clause;
- The possibility of concluding an agreement without such a clause with another person;
- Knowledge of the existence of the clause.[196]

[212] When free and open source software is involved, certain elements are propitious to judicial intervention. First, the licensor clearly has a monopoly on bargaining power. In this regard, the mere designation as a contract of adhesion may sometimes suffice to invalidate liability exemptions.[197] Second, it is important to bear in mind that all software licences, whether free and open source or proprietary, contain similar clauses. Thus, the licensee often has no option but to acquiesce to this regime. Third, when the software is commercially marketed, that absence of contractual protection may be invalidated on the basis of the amount paid. Fourth, free and open source licences that seek to exclude all forms of civil liability out of hand, by means of a single clause, may be at a disadvantage. Indeed, this technique

[194] *Québec Civil Code*, *op. cit.*, footnote 11, sect. 1474.

[195] M. VIVANT et al., *op. cit.*, footnote 157, no. 1129, p. 662.

[196] David SLEE, "Liability for Information Provision", (1992) 3 *The Law Librarian* 155, 157.

[197] *Québec civil code*, *op. cit.*, footnote 11, sect. 1437.

may be deemed abusive, since it would be possible to use specific clauses for each type of risk envisaged.[198]

[213] Conversely, several arguments can be advanced to support exemption clauses of free and open source licences. First, an alternative exist to the acceptation of these risks, since the licensee always has the option of taking out insurance, or of contracting with a service supplier who warrants the computerized solution he implements. Also, it is essential to account for the extent of the rights granted by the developers of free and open source software in exchange for this exemption. Since the licensee obtains authorization to copy, modify, and redistribute the software, the contractual balance appear to be satisfied. Furthermore, since the exemption clause is always prominent and written in clear language, the licensee may find it difficult to claim that he was unaware of it when he agreed to the terms of the licence. Finally, it should be kept in mind that, most of the time, licensors give of their time and knowledge. A very large number of people have the means to become involved in the development of free and open source software, keeping production costs low, provided they are excluded from contractual liability.[199]

[214] Overall, it appears that the balance of probabilities leans in favour of the validity of exemption clauses in the case of free and open source software distributed without cost. In the matter of free and open source software distributed for a remuneration, the solution will probably be different in each particular case, in light of the specific circumstances and the amount paid by the licensee.

8.4. Consumer Protection

[215] In addition to general contract law, Quebec law include a legislative text specifically designed to protect consumers: The *Consumer Protection Act*.[200] This law can have significant effects on contracts subjected to it. Its application to free and open source licences first poses the problem designating the parties. Consideration must be given to determining whether the licensee corresponds to the given definition of consumer and the licensor to that of a merchant.

[216] In general, the licensee can invoke this law if he is an individual and the licence contract has no direct link with his professional activities. Indeed, the law

[198] D. SLEE, *loc. cit.*, footnote 64, 158.

[199] *Robert W. GOMULKIEWICZ, "How Copyleft Uses License Rights to Succeed in the Open Source Software Revolution and the Implications for Article 2b˝, (1999) 36 Hous. L. Rev. 179, 192.*

[200] Québec *Consumer Protection Act*, op cit., footnote 79.

specifies that a consumer is "a natural person, except a merchant who obtains goods or services for the purposes of his business."[201] Thus, this definition peremptorily excludes the public administration from any application of the *Consumer Protection Act*. Similarly, if a civil servant, being a natural person, were to contract for a free and open source licence within the context of his work, the direct link between the contract and his professional activity would deprive him of this protection.

[217] In the specific context of free and open source software, the designation of the licensor as a merchant must also be assessed. Indeed, a large portion of free and open source software developers are clearly not merchants, since they are not performing commercial acts within the context of their profession. Though frequently professionals in the computer industry, they are usually acting without any profit motive and outside of the framework of their principal activity. *A priori*, the *Consumer Protection Act* should not be applied to such a situation. Conversely, firms whose business model is based on the distribution of free and open source software can certainly be considered as merchants, making them subject to this Act.

[218] If the applicability of this Act is accepted, consumer protection law steps in to protect the licensee. This protection is particularly useful in the matter of abusive clauses.[202] In this matter, section 8 stipulates that:

> "The consumer may demand the nullity of a contract or a reduction in his obligations thereunder where the disproportion between the respective obligations of the parties is so great as to amount to exploitation of the consumer or where the obligation of the consumer is excessive, harsh or unconscionable."

[219] This provision could potentially be applied by a court to a disclaimer included in a free and open source licence. This could also be the case if particularly restrictive obligations were imposed on a licensee by a given licence.

[220] Among the other protections provided for by the law, we specifically find supplemental warranties and provisions designed to reinforce existing warranties. These additional protections are equivalent to the warranty of merchantability existing in common law jurisdictions.[203] Consequently, as long as the *Consumer Protection Act* applies, a licensor could never be exonerated from the warranty

---

[201]Québec *Consumer Protection Act*, op cit., footnote 79, sect. 1.

[202]Pierre-Emmanuel MOYSE, Vincent GAUTRAIS, "Droit des auteurs et droit de la consommation dans le cyberespace: la relation auteur/utilisateur", (1996) *Léger Robic Richard*, source: <http://www.robic.ca/publications/Pdf/070-PEM.pdf>.

[203]Québec *Consumer Protection Act*, op cit., footnote 79, sect. 37 and 38.

against latent defects (sect. 53) or liberated from the consequences of his own act (sect. 10). He must also warrant that the good is free "from every charge or encumbrance in favour of a third person" (sect. 36), which corresponds to the warranty against eviction from third parties. The law also expressly provides a warranty that the good corresponds to its description and any advertising on its behalf (sect. 40 and 41). Furthermore, it specifies that any written or verbal claim, and any written warranty, pertaining to the software constitutes an express warranty that is binding against the licensor (sect. 42 and 43). These provisions all being of public order nature, exemption clauses are insufficient to override them. In conclusion, if the *Consumer Protection Act* is applicable to a free and open source licence, the civil liability of the licensor is greatly expanded and, correspondingly, the protection of the licensee is greatly enhanced.

[221] *A priori*, it appears that, at the very least, this law should protect licensees having paid to acquire free and open source software. This does not create any problems, to the extent that the agreements are designated as sales or lease, since these are covered by the traditional scope of this legislation. Conversely, if they are deemed to be innominate contracts, it is less certain that the consumer will be able to benefit from them. Nonetheless, most likely the courts will correct this situation by treating licence contracts as consummer contracts anyway. Tit can be explained by the broad and liberal interpretation that must be given to the *Consumer Protection Act*, which has the "vocation of encompassing all contracts entered into by consumers in the mass market. (author's translation)"[204]

[222] It is unlikely that free and open source software distributed without cost will receive the same treatment. Assuming that these licences are construed as loans, application of the consumer protection regime is automatically precluded, at it specifically targets sale and lease contracts. In addition, the object of this law is to protect the economic interests of a weaker party, the consumer, against those of a stronger party, the merchant. It would be paradoxical if it were to be applied in a situation in which the consumer receives many rights from a merchant who renounces to any compensation.

[204]Vincent GAUTRAIS, Ejan MACKAAY, "Les contrats informatiques", in Denys-Claude LAMONTAGNE, *Contrats spéciaux*, Éditions Yvon Blais, Cowansville, 2001, p. 279.

9. Recommendations

[223] While no legal objection impedes the use of free and open source software by the Government of Québec, similarly as for proprietary software, certain specific legal risks do arise. Therefore, establishing a technology strategy centred on free and open source software by the Government of Québec requires implementing some preventive measures to reduce and manage these risks. These measures, far from representing a burden on the Québec administration, have the objective of facilitating the integration of the legal documents that are free and open source licences while ensuring long-term legal security.

9.1. Accounting for pre-existing contractual agreements during the elaboration of migration plans

[224] The large scale transition from one technological environment to another requires the elaboration of detailed road maps in order to ensure that all factors likely to have an impact on the success of the operation are taken into account. These migration plans usually target the services and functions affected by the transition, chosen of replacement applications, and human factors, such as training requirements. However, when this operation also involves abandoning proprietary solutions for which licence and service contracts have been concluded, this additional element should be taken into consideration.

[225] Depending on the contractual agreements already concluded by the government, it may be binded to some companies for several years. Furthermore, it may be the case that specific proprietary software must be retained in order to honour commitments made by the public administration in various contractual framework with third parties. In either of these cases, a hasty migration to free and open source technologies may involve the government liability. Ultimately, a poor coordination of technology contracts may result in an escalation of migration costs, eradicating at hte same time some part of the savings that were sought from the transition to free and open source software. Under these circumstances, any migration schedule must account for pre-existing contractual agreements.

9.2. Elaborating a procedure for accepting free and open source licences

[226] The use of free and open source software by the government of Québec necessarily implies that it must, in its role as licensee, agree to comply with the terms of the licences that characterize it. Since these terms vary, as do the legal

consequences arising from them, it is far from certain that all free and open source licences will prove acceptable to the public administration. Consequently, it is essential to identify the free and open source licences preferred by the administration and to establish a process for ensuring that these choices are respected.

[227] Furthermore, acceptance of a free and open source licence must also account for the context that frames the government's technological needs. Any time that the use of free and open source software is required, the terms of its licence should be evaluated from the perspective of the objectives sought by the government. At this stage, the following factors should be considered:

- Are the terms of the licence clear? Are any ambiguities likely to result in divergent interpretations?
- If any modification and redistribution of the software is intended, are the reciprocity requirements of the licence acceptable?
- If any modification and redistribution of the software is intended, is its licence compatible with those of the other software with which it is to be integrated?
- Are any additional rights to those provided by the licence required for effective use of the software?
- Are the exemption clauses included in the licence are acceptable, given the use for which the software is intended?
- Are there any clauses precluding the application of Québec law and the jurisdiction of Québec courts?
- What is the likelihood that the licence termination clauses will find application, given the use for which the software is intended?

[228] In addition to accounting for these various elements, it is in the interest of the Government of Québec to require that the software's authors provide a written waiver of their moral rights whenever possible. Clearly, this is only an option if the government has directly contracted with a service provider having developed the software itself. In all other cases, the public administration will remains subject to software authors' right of review, springing from their right to integrity.

[229] Establishing a mechanism to ensure that these elementary rules of prudence are followed should minimize the legal risks arising from the use of free and open source software by the Government of Québec.

9.3. Elaborating a procedure for granting free and open source licences

[230] In light of the powers it bestows on licensees, the use of free and open source licences also provide the Government of Québec with the possibility of participating in the modification and distribution of the software. This attitude is perfectly consistent with the general practice in this field, which associates the exercise of these powers with a duty to make the community benefit from any constructive work completed. Similarly, any software projects initiated by the public administration could be governed by free and open source licences. In both cases, the Government of Québec switches from being a simple user (licensee) to being a developer (licensor).

[231] There can be no doubt that this change in status requires the definition and implementation of a procedure for validating the legal status of software thus released by the government. While establishing such a mechanism does not necessarily involve implementing complex measures, it does require evaluating certain issues that fall outside of the scope of this paper.

[232] Among other things, this procedure must necessarily account for the various legal frameworks likely to apply to the different developer types participating in the creation of software for the public administration. Indeed, the computer code generated by permanent civil servants is not necessarily subject to the same rules as code supplied by service providers or even by independant workers recruited for specific projects. Furthermore, it is far from clear that software created by the public administration would always benefit from all the protections afforded by the copyright regime.

[233] An even more fundamental question addresses the identification of what government software should be released under a free and open source licence, and what code must absolutely remain inaccessible. For example, two situations are clearly incompatible with adopting a free and open source licence:

- the software must remain confidential;

- dual grants are inappropriate and the government intends to collect royalties from its ownership of intellectual property rights over the software.[205]

[234] Once the affected software has been identified, it remains to select the licence(s) to protect the government's creations. In this matter, it appears important to account for the level of protection that code owned by the Government of Québec will receive abroad.

[235] Finally, any position taken in favour of releasing government software under free and open source licences will raise questions regarding how the intellectual property rights over other protected works belonging to the government are exploited. Indeed, the restrictive conditions placed on the reproduction and reuse of various administrative documents will become incongruous. Under these circumstances, it appears appropriate to envisage adopting an equivalent model, as proposed by the Creative Commons movement,[206] for a large part of the digital content produced by the government.

[236] A more in-depth examinating of the role of the Government of Québec as a licensor should be performed to answer all these questions. This would allow establishing and enforcing a process for granting free and open source licences that is clear and adapted to the Québec context.

9.4. Adopting a consistent policy regarding software patents

[237] Given the official position of Canada's federal authorities in the matter of software patentability, the problems associated with software patents are not likely to have an impact on Québec's free and open source software strategy, at least in the short term. Nonetheless, in consideration of the ongoing barrage of requests for software patents to which the Patent Office is subjected, it must remain resolute to ensure adequate protection to creators and users of free and open source software in Québec.

[238] The importance of maintaining this position is accentuated in light of the lobbying conducted by the U.S. government in favour of software patents. A study conducted in 2003 and funded by Industry Canada has revealed that Canada is

[205]Brendan SCOTT, "Why the State Should Release Its Software as Open Source", (2004) *Open Source Law*, source : <http://www.opensourcelaw.biz/papers/BScottGovAccessRegimes040519.pdf>.

[206]CREATIVE COMMONS, "Learn More about Creative Commons", (2005) *Creative Commons*, source : <http://creativecommons.org/learnmore>.

already responding to U.S. pressures and is currently assessing the implications of software patents.[207] Far from supporting the U.S. position, the study *Review of Software Patent Issues* emphasizes the danger to the development growth of free and open source software in Canada inherent in it.

[239] Any move to harmonize Canadian law with this position would increase the legal risk associated with the use of free and open source software by the Government of Québec, which would then be exposed to litigation from all software patent owners who may have a claim related to infringements committed during the development of any free and open source software used by the public administration. Undoubtedly, such a scenario has the potential to be very onerous for the Government of Quebec.

[240] Consequently, any support from Québec for the development and use of free and open source software requires that the government pressure federal authorities to oppose software patents. Any other position would be inconsistent.

9.5. Adopting mechanisms for the management of legal risk

[241] Even if all the aforementioned measures are implemented, the use of free and open source software will continue to involves certain legal risks. This is no different, incidentally, than with proprietary software. In the latter case, for example, the firm owning the rights over the software may go bankrupt, sell these rights to a third party, or simply unilaterally cease supporting or developing the software.

[242] In the case of free and open source software, the bulk of the risk to the Government of Québec springs from the high level of protection afforded to the licensor. In many cases, the licensor is even difficult to identify.[208] Moreover, a number of licensors are simply not compellable, either because they live abroad or because their financial capacity is limited. Under these circumstances, the public administration, as a licensee, must be ready to assume a large proportion of the risk itself.

[243] Two solutions should be implemented by the Government of Québec to lighten the burden resulting from this situation. The first relates to the involvement

[207]R. McORMOND, *loc cit.*, footnote **Erreur ! Signet non défini.**.

[208]Valérie SEDALLIAN, "Garanties et responsabilité dans les logiciels libres", (2002) 152 *Lamy droit de l'informatique et des réseaux* 1, 7.

of local service providers. The second consists of taking out insurance for the residual legal risks.

[244] One of the primary benefits of free and open source software is the possibility to integrate them into customized technological solutions. Thus, in many cases, the use of free and open source software by the government will originate with a local service provider having based his own applications on it. This approach features the dual benefit of promoting the local software industry and, especially, to guarantee the effectiveness of the a minimal warranties protecting the public administration. Even though no Québec service provider will undertake by himself to warrant free and open source software developed by others, the government's bargaining power should allow it to reach a much greater contractual balance than what is specified in most of the examined licences. In addition, simply involving a compellable intermediary in the relationship will results in a more efficient management of risks.

[245] Furthermore, by involving service providers as often as possible, the government also facilitates the tracking of the entry of free and open source software into the public administration. Thus, the legal validation that will eventually be necessary at the time of it modification and redistribution will also be simplified.

[246] In the event that the level of risk needs to be reduced even more, the Government of Québec always has the option of taking out insurance to that effect. Thus, it may cover itself against claims from third parties pretenting to own rigths in any of the free and open source software used by the publicadministration. In response to growing demand, several international service providers, such as Novell, have even created compensation funds in the aftermath of the SCO litigation. However, since these funds are all specific to a given technology, and their coverage generally limited to copyright infringements, broader coverage should be envisaged.[209]

9.6. Renouncing to the adoption of a free and open source software promotion law

[247] In order to stress their position and compel the civil service to use free and open source software, some governments have opted for a legislative solution. Thus, a law obliges civil servants in the Australian capital state to consider free and open

[209] Daniel EGGER, "Why the Linux Community Needs Open Source Insurance", (2004) *LinuxWorld*, source: <http://www.linuxworld.com/story/44110.htm?DE=1>.

source software when buying software while at work, and to avoid software that relies on proprietary standards.[210] However, this radical approach entails several problems, and the Government of Québec should avoid emulate it.

[248] In practice, local firms that develop proprietary software would be hurt the most if this type of legislation were passed. Indeed, it would make a large part of their market inaccessible to them. However, there is no reason why the use of proprietary software should be completely abandoned, even when a free and open source alternative exists. A variety of factors may make it advisable for the public administration to continue using a proprietary solution. One example may be the availability of sufficient trainers.

[249] From a legal perspective, any move to favor free and open source software over other types of software may run a risk of violating commitments made by the Government of Canada in the framework of agreements concluded within the World Trade Organization (WTO). Under these agreements, purchases by member states must comply with general principles of non-discrimination elaborated in the GATT.[211] As a consequence, the choice of software used by the public administration must be made exclusively on the basis of business considerations, including price, quality, and availability. Imposing the use of free and open source software would thus put the Government of Québec in the position of discriminating against foreign software publishers. Even if the use of proprietary software would not be totally proscribed by the law, the mere fact of establishing strict criteria, such as requiring the use of open standards, may lead to equivalent consequences.[212] Under such circumstances, other members states of the WTO (with the United States being the principal interested party) would be entitled to require that the Government of Canada correct the situation.

[250] Similarly, the *North American Free Trade Agreement* (NAFTA) stipulates that entities in Canada may not "prepare, adopt or apply any technical specification with the purpose or the effect of creating unnecessary obstacles to trade."[213] As a result, any legislated preference for free and open source software could be contested by

---

[210] *Government Procurement (Principles) Guideline Amendment Act* 2003, op cit., foot note 5.

[211] *The General Agreement on Tariffs and Trade* (1947), World Trade Organization, source: <http://www.wto.org/english/docs_e/legal_e/gatt47_01_e.htm>.

[212] Shanker A. SINGHAM, D. Daniel SOKOL, "Public Sector Restraints: Behind-the-Border Trade Barriers", (2004) 39 *Tex. Int'l L. J.* 625, 639.

[213] *North American Free Trade Agreement*, (2005), NAFTA Secretariat, article 1007, source: <http://www.nafta-sec-alena.org/DefaultSite/index_e.aspx?DetailID=136#A1007>.

North American software publishers wishing to participate in a call for tenders issued by the Government of Québec. This is all the more relevant since, in the opposite direction, the Canadian International Trade Tribunal already invalidated proprietary specifications adopted by the federal government and ruled in favour of a firm whose business model was based on free and open source software.[214]

[251] All in all, it is important that the integration of free and open source software into the Government of Québec's technology strategy does not assume the form of a legal constraint. Conditions propitious for reaching that goal can be established without legislation, relying primarily on sensitization and education.

[214] *P&L Communications Inc.*, File No.: PR-2000-059, (2000) CITT, source: <http://www.citt-tcce.gc.ca/procure/determin/pr2a059_e.asp>.

10. Conclusion

[252] Overall, Québec law appears to be up to the challenge of effectively addressing the various legal issues underlying the use of free and open source software.

[253] First of all, no legal rule restrict the validity of free and open source licences in Québec, despite that fact that few of them were designed with the Québec legal system in mind. In fact, the requirement of the *Copyright Act* making the validity of licences contingent on the existence of a written signature is the only one liable to raise serious difficulties in this matter. Nonetheless, the rules governing electronic signatures that are applicable in Québec should suffice to validate the effectiveness of most free and open source licences. As with the issue relating to the manifestation of the licensee's consent, the answer to this question depend on the facts in each case, which require a specific analysis.

[254] Also, there is no doubt that Canadian copyright law affords effective protection to the licensors of free and open source software. In fact, the authors receive rights that even exceed those they are granted under U.S. law — moral rights to paternity and integrity. The exercise of this latter represents a threat to the unfettered evolution of free and open source software because of the permanent right of review it confers on authors. Nonetheless, the primary legal risk incurred by the users of free and open source software resides in weak links that may be present higher in the chain of title. Unfortunately, there are no measures that can pre-empt such claims based on intellectual property rights that may be asserted by third parties against the users of free and open source software.

[255] Notwithstanding these difficulties, usage of free and open source software within Québec simply requires paying a minimum of attention to some of its specificities. The first of these relates to the compatibility between various free and open source licences, which is far from always certain. The second concerns software patents, the acceptance of which could result in real legal risks in Canada .

[256] Finally, application of the terms of free and open source licences under Quebec law also ensures an effective protection to licensors. However, this protection is not absolute, since under some circumstances exemption clauses of free and open source licences may be void. This is the case, notably, when the acts of developers constitute a gross fault or negligence, when they are committed with the intention of causing harm, or when the law prohibits contractual exemptions for

a specific type of warranty or damage. Developers who distribute free and open source software commercially face considerably higher risks, since these circumstances argue for limiting the effects of exemption clauses, while the protections of the *Consumer Protection Act* can be brought to bear against them.

[257] In light of these assertions, the Government of Québec can easily adopt free and open source software, at least as far as the law is concerned, provided some measures are implemented. Overall, these initiatives seek to reduce the legal risks identified above as much as possible, and to allow the management of those that cannot be completely eliminated. The cornerstone of this strategy is certainly the establishment of procedures for legal validation upstream from the implementation and the eventual modification and redistribution of free and open source software by the public administration. By ensuring that lawyers participate in the acceptation process of free and open source licences, the government can reduce its chances of finding itself in an irregular situation resulting in the involvement of its liability.

[258] Given the current environment, in which Quebec's software industry is showing a burgeoning interest in free and open source software, the time seems particularly propitious for the Government of Québec to envisage integrating it into its technology strategy.

12. Caveats

[259] The authors invested all of their knowledge and experience in writing this study. Nonetheless, some errors and omissions will remain. In addition, the opinions expressed in this document are those of the authors, it is possible that others may hold diverging views.

[260] The information provided in the framework of this study should not be construed as legal guidance, in the same way that this study should not be interpreted as legal advice or treated as a substitute for obtaining legal advice from a competent legal advisor.

[261] The information provided in this study are considered up-to-date as of January 1, 2005.

13. Selected bibliography

Legislation

North American Free Trade Agreement, (2005) NAFTA, source : <http://www.nafta-sec-alena.org/DefaultSite/index_f.aspx?DetailID=305>

The General Agreement on Tariffs and Trade , (1947) World Trade Organization, source : <http://www.wto.org/french/docs_f/legal_f/gatt47_01_f.htm>

Berne Convention for the Protection of Literary and Artistic Works, (2005) Juris International, source : <http://www.jurisint.org/pub/01/fr/147.htm>

Quebec Civil Code, Q.A. 1991, c. 64, source : <http://www.canlii.org/qc/legis/loi/ccq/index.html>

An Act to Establish a Legal Framework for Information Technology, Q.A., c. C-1.1, source : <http://www.canlii.org/qc/legis/loi/c-1.1/index.html>

Constitutional Act of 1867, source : <http://www.canlii.org/ca/const_fr/const1867.html>

Personnal Information Protection and Electronic Documents Act, C.A. (2000), c. 5, source : <http://www.canlii.org/ca/loi/p-8.6/>

Consummer Protection Act, Q.A., 1977, c. P-40.1, source : <http://www.canlii.org/qc/legis/loi/p-40.1/index.html>

Copyright Act, C.A. (1985), c. C-42, source : <http://www.canlii.org/ca/loi/c-42/>

Patent Act, C.A. (1985), c. P-4, source : <http://www.canlii.org/ca/loi/p-4/>

Jurisprudence

Apple Computer c. *Mackintosh Computers Ltd.*, [1987] 1 F.C.C. 173

CCH Canadienne Ltée c. *Barreau du Haut-Canada*, 2004 SCC 13, source : <http://www.canlii.org/ca/jug/csc/2004/2004csc13.html>

Delrina Corp. v. Triolet Systems Inc. (1993) 47 C.P.R. (3rd) 1

Diffusion YFB Inc. c. Les Disques Gamma (Québec) Ltée, (1999) C.S. 500-05-047570-997

Galerie d'art du Petit Champlain Inc. c. Théberge, 2002 SCC 34, source : <http://www.canlii.org/ca/jug/csc/2002/2002csc34.html>

IBM Corp. c. *Ordinateurs Spirales Inc.*, [1985] 1 F.C. 190

Informatique L.G.A. Inc. c. *Compagnie d'arrimage de Québec Ltée,* [1991] R.J.Q. 1767 (C.Q.)

Kanitz v. *Rogers Cable Inc.*, (2002) 58 O.R. (3rd) 299, source : <http://www.dww.com/decisions/kanitz_v_rogers_cable_inc.pdf>

Les Amusements Wiltron Inc. c. *Mainville*, [1991] R.J.Q. 1930 (C.S.)

Marquis c. *DKL Technologies Inc.,* (1989) 24 C.I.P.R. 289 (C.S.Q.)

Olier, Grisé & Cie Ltée c. *Équipements de bureau Maskoutan Inc.,* [1985] C.S. 680

P&L Communications Inc., Dossier no PR-2000-059, (2000) TCCE, source : <http://www.citt-tcce.gc.ca/procure/determin/pr2a059_f.asp>

Prism Hospital Software Inc. c. *Hospital Medical Records Institute et al.*, (1994) 57 C.P.R. (3d) 129 (B.C.S.C.)

Re Application for Patent of Motorola Inc., (1999) 86 CPR (3d) 71 et 76.

Rudder v. *Microsoft Corp.*, (1999) O.S.C.J., source : <http://www2.droit.umontreal.ca/cours/ecommerce/_textes/rudder.doc>

Schlumberger Canada Ltd. c. Commissaire des brevets, (1981) 56 C.P.R. (2d) 204 (C.A.F.)

Sillons Le Disquaire Inc. c. Datagil Informatique Inc., (1998) J.E. 98-1148 (C.Q.)

Télé-direct (Publications) Inc. c. American Business Information, Inc., [1998] 2 F.C. 22, source : <http://www.canlii.org/ca/jug/caf/1997/1997caf10177.html>

Tremblay c. *Nguyen,* (1997) 24 C.P.R. 289

Doctrine

CARON, C., « Les licences de logiciel dits 'libres' à l'épreuve du droit d'auteur français », (2003) 23 *Dalloz* 1556

CARRIÈRE, L., « Protection du logiciel et autres oeuvres originales en vertu de la Loi sur le droit d'auteur et conventions usuelles en ce domaine » (1996) *Léger Robic Richard*, source : <http://www.robic.ca/PublicationsFr.shtml>

CLÉMENT-FONTAINE, M., « La licence publique générale GNU », (1999) *Juriscom*, source : <http://www.juriscom.net/uni/mem/08/presentation.htm>

DIBONA, OCKMAN et STONE (dir.), *Open Sources: Voices from the Open Source Revolution*, O'Reilly, 1999, source : <http://www.oreilly.com/catalog/opensources/book/raymond.html>

GAUTRAIS, V., « Les contrats de cyberconsommation sont presque tous illégaux! » (2005) à être publié dans la revue du Notariat

GAUTRAIS, V., MACKAAY, E., « Les contrats informatiques », dans Denys-Claude LAMONTAGNE, *Contrats spéciaux*, Éditions Yvon Blais, Cowansville, 2001, p. 279

GOETTSCH, K. D., " Sco Group v. IBM : the future of open-source software " (2003) *U. Ill. J.L. Tech. & Pol'y* 581

GRAVELLE, L.-P., « De la vie aux ordinateurs – développements récents en matière de brevetabilité des organismes vivants, des méthodes de traitement et des technologies informatiques », (2000) 138 *Développements récents en propriété intellectuelle* 104, source : <http://www.robic.ca/publications/Pdf/254-LPG.pdf>

KENNEDY, D. M., " A Primer on Open Source Licensing Legal Issues : Copyright, Copyleft and Copyfuture ¨, (2001) 20 *St. Louis U. Pub. L. Rev.* 345

LESSIG, L., *Free Culture*, Penguin Press, New York, 2004

LESSIG, L., *The Future of Ideas: The Fate of the Commons in a Connected World*, Vintage, New York, 2002

LE TOURNEAU, P., *Théorie et pratique des contrats informatiques*, Éditions Dalloz, Paris, 2000

LUCAS, A., DEVÈZE, J., FRAYSSINET, J., « Droit de l'informatique et de l'internet », PUF, *coll. Thémis Droit privé*, n° 760

MADISON, M. J., "Reconstructing the Software License", (2003) 35 *Loy. U. Chi. L.J.* 275

MACKAAY, E., « Le marché du progiciel – licence ou vente? », (1994) 6 *Cah. Prop. Int.* 401

MCORMOND, R., "A Review of Software Patent Issues", (2003) *Digital Copyrigth Canada*, source : <http://www.flora.ca/patent2003/>

MOKANOV, I., « La teneur du standard de fiabilité des moyens électroniques de signature », (2004) 9 *Lex Electronica* 1, source : <http://www.lex-electronica.org/articles/v9-1/mokanov.htm>

MOYSE, P.-E., GAUTRAIS, V., « Droit des auteurs et droit de la consommation dans le cyberespace : la relation auteur/utilisateur », (1996) *Léger Robic Richard*, source : <http://www.robic.ca/publications/Pdf/070-PEM.pdf>

NADAN, C. H., " Open Source Licensing Virus or Virtue " (2002) 10 *Tex. Intell. Prop. L.J.* 349

PRICE WATERHOUSE COOPERS, Rethinking the European ICT Agenda : Ten ICT-Breakthroughs for Reaching Lisbon Goals, (2004) *Price Waterhouse Coopers*, source : <http://www.pwc.com.nyud.net:8090/Extweb/pwcpublications.nsf/docid/EC6DE73A846581CE80256EFD002E41FB/$file/pwc_rethinking_european_ict_agenda.pdf>

RAYMOND, E. S., " The Cathedral and the Bazaar ", (1998) 3 *First Monday*, source : <http://www.firstmonday.org/issues/issue3_3/raymond/index.html>

ROJINSKY, C., GRYNBAUM, V., *Les licences libres et le droit français*, Propriété intellectuelles, juill. 2002/4

ROSEN, L., *Open Source Licensing : Software Freedom and Intellectual Property Law*, Prentice Hall PTR, Upper Saddle River, 2004

SAPP, N., « La contrefaçon en matière de logiciel : où en sommes-nous? », (2000) 138 *Développements récents en propriété intellectuelle* 161

SCOTT, B., " Why the State Should Release Its Software as Open Source ", (2004) *Open Source Law*, source : <http://www.opensourcelaw.biz/papers/BScottGovAccessRegimes040519.pdf>

SEDALLIAN, V., « Garanties et responsabilité dans les logiciels libres », (2002) 152 *Lamy droit de l'informatique et des réseaux* 1

SMETS-SOLANES J.-P., FAUCON, B., *Logiciels Libres : Liberté, Egalite, Business, Freepatents*, Edispher, 1999

ST. LAURENT, A. M., *Understanding Open Source & Free Software Licensing*, O'Reilly, Sebastopol, 2004

STRASSER, M., "A New Paradigm in Intellectual Property Law? The Case Against Open Sources", (2001) *Stan. Tech. L. Rev.* 4, source : <http://stlr.stanford.edu/STLR/Articles/01_STLR_4/article.htm>

TAMARO, N., *The 2004 Annotated Copyright Act*, Thomson Carswell, Toronto, 2004

VAVER, D., *Copyright Law*, Irwin Law, Toronto, 2000

VETTER, G. R., "The Collaborative Integrity of Open-Source Software", (2004) *Utah L. Rev.* 563

VIVANT, M., et autres, *Lamy droit de l'informatique et des réseaux*, Lamy, Paris, 2001

WEBBINK, M. H., " Open Source Software – Bridging the Chasm ", (2002) 691 *PLI/Pat* 663

WHEELER, D. A., "Make Your Open Source Software GPL-Compatible. Or Else.", (2005) *DWheeler*, source : <http://www.dwheeler.com/essays/gpl-compatible.html>

WILLIAMS, S., *Free as in Freedom : Richard Stallman's Crusade for Free Software*, O'Reilly, Sebastopol, 2002