Reverse Engineering Computer Programs Under Canadian Copyright Law

by Sunny Handa
Faculty of Law
McGill University, Montreal
September, 1994

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements of the degree of Master of Laws (LL.M.) at McGill University, Montreal, Canada.

ISBN 0-612-05493-4

Canada

Name Mr. Sunny P. Harida

Dissertation Abstracts International is arranged by broad, general subject categories. Please select the one subject which most nearly describes the content of your dissertation. Enter the corresponding four-digit code in the spaces provided.

Reverse Engineering Computer Programs Under Canadian Copyright Law
SUBJECT TERM

0398 SUBJECT CODE

U·M·I

Subject Categories

THE HUMANITIES AND SOCIAL SCIENCES

COMMUNICATIONS AND THE ARTS
Architecture 0729
Art History 0377
Cinema 0900
Dance 0378
Fine Arts 0357
Information Science 0723
Journalism 0391
Library Science 0399
Mass Communications 0708
Music 0413
Speech Communication 0459
Theater 0465

EDUCATION
General 0515
Administration 0514
Adult and Continuing 0516
Agricultural 0517
Art 0273
Bilingual and Multicultural 0282
Business 0688
Community College 0275
Curriculum and Instruction 0727
Early Childhood 0518
Elementary 0524
Finance 0277
Guidance and Counseling 0519
Health 0680
Higher 0745
History of 0520
Home Economics 0278
Industrial 0521
Language and Literature 0279
Mathematics 0280
Music 0522
Philosophy of 0998
Physical 0523
Psychology 0525
Reading 0535
Religious 0527
Sciences 0714
Secondary 0533
Social Sciences 0534
Sociology of 0340
Special 0529
Teacher Training 0530
Technology 0710
Tests and Measurements 0288
Vocational 0747

LANGUAGE, LITERATURE AND LINGUISTICS
Language
General 0679
Ancient 0289
Linguistics 0290
Modern 0291
Literature
General 0401
Classical 0294
Comparative 0295
Medieval 0297
Modern 0298
African 0316
American 0591
Asian 0305
Canadian (English) 0352
Canadian (French) 0355
English 0593
Germanic 0311
Latin American 0312
Middle Eastern 0315
Romance 0313
Slavic and East European 0314

PHILOSOPHY, RELIGION AND THEOLOGY
Philosophy 0422
Religion
General 0318
Biblical Studies 0321
Clergy 0319
History of 0320
Philosophy of 0322
Theology 0469

SOCIAL SCIENCES
American Studies 0323
Anthropology
Archaeology 0324
Cultural 0326
Physical 0327
Business Administration
General 0310
Accounting 0272
Banking 0770
Management 0454
Marketing 0338
Canadian Studies 0385
Economics
General 0501
Agricultural 0503
Commerce-Business 0505
Finance 0508
History 0509
Labor 0510
Theory 0511
Folklore 0358
Geography 0366
Gerontology 0351
History
General 0578
Ancient 0579
Medieval 0581
Modern 0582
Black 0328
African 0331
Asia, Australia and Oceania 0332
Canadian 0334
European 0335
Latin American 0336
Middle Eastern 0333
United States 0337
History of Science 0585
Law 0398
Political Science
General 0615
International Law and Relations 0616
Public Administration 0617
Recreation 0814
Social Work 0452
Sociology
General 0626
Criminology and Penology 0627
Demography 0938
Ethnic and Racial Studies 0631
Individual and Family Studies 0628
Industrial and Labor Relations 0629
Public and Social Welfare 0630
Social Structure and Development 0700
Theory and Methods 0344
Transportation 0709
Urban and Regional Planning 0999
Women's Studies 0453

THE SCIENCES AND ENGINEERING

BIOLOGICAL SCIENCES
Agriculture
General 0473
Agronomy 0285
Animal Culture and Nutrition 0475
Animal Pathology 0476
Food Science and Technology 0359
Forestry and Wildlife 0478
Plant Culture 0479
Plant Pathology 0480
Plant Physiology 0817
Range Management 0777
Wood Technology 0746
Biology
General 0306
Anatomy 0287
Biostatistics 0308
Botany 0309
Cell 0379
Ecology 0329
Entomology 0353
Genetics 0369
Limnology 0793
Microbiology 0410
Molecular 0307
Neuroscience 0317
Oceanography 0416
Physiology 0433
Radiation 0821
Veterinary Science 0778
Zoology 0472
Biophysics
General 0786
Medical 0760

EARTH SCIENCES
Biogeochemistry 0425
Geochemistry 0996
Geodesy 0370
Geology 0372
Geophysics 0373
Hydrology 0388
Mineralogy 0411
Paleobotany 0345
Paleoecology 0426
Paleontology 0418
Paleozoology 0985
Palynology 0427
Physical Geography 0368
Physical Oceanography 0415

HEALTH AND ENVIRONMENTAL SCIENCES
Environmental Sciences 0768
Health Sciences
General 0566
Audiology 0300
Chemotherapy 0992
Dentistry 0567
Education 0350
Hospital Management 0769
Human Development 0758
Immunology 0982
Medicine and Surgery 0564
Mental Health 0347
Nursing 0569
Nutrition 0570
Obstetrics and Gynecology 0380
Occupational Health and Therapy 0354
Ophthalmology 0381
Pathology 0571
Pharmacology 0419
Pharmacy 0572
Physical Therapy 0382
Public Health 0573
Radiology 0574
Recreation 0575
Speech Pathology 0460
Toxicology 0383
Home Economics 0386

PHYSICAL SCIENCES
Pure Sciences
Chemistry
General 0485
Agricultural 0749
Analytical 0486
Biochemistry 0487
Inorganic 0488
Nuclear 0738
Organic 0490
Pharmaceutical 0491
Physical 0494
Polymer 0495
Radiation 0754
Mathematics 0405
Physics
General 0605
Acoustics 0986
Astronomy and Astrophysics 0606
Atmospheric Science 0608
Atomic 0748
Electronics and Electricity 0607
Elementary Particles and High Energy 0798
Fluid and Plasma 0759
Molecular 0609
Nuclear 0610
Optics 0752
Radiation 0756
Solid State 0611
Statistics 0463

Applied Sciences
Applied Mechanics 0346
Computer Science 0984
Engineering
General 0537
Aerospace 0538
Agricultural 0539
Automotive 0540
Biomedical 0541
Chemical 0542
Civil 0543
Electronics and Electrical 0544
Heat and Thermodynamics 0348
Hydraulic 0545
Industrial 0546
Marine 0547
Materials Science 0794
Mechanical 0548
Metallurgy 0743
Mining 0551
Nuclear 0552
Packaging 0549
Petroleum 0765
Sanitary and Municipal 0554
System Science 0790
Geotechnology 0428
Operations Research 0796
Plastics Technology 0795
Textile Technology 0994

PSYCHOLOGY
General 0621
Behavioral 0384
Clinical 0622
Developmental 0620
Experimental 0623
Industrial 0624
Personality 0625
Physiological 0989
Psychobiology 0349
Psychometrics 0632
Social 0451

McGill University
FACULTY OF GRADUATE STUDIES AND RESEARCH

AUTHOR'S NAME: Mr. Sunny Handa

DEPARTMENT: LAW (I.C.L.) DEGREE SOUGHT: LL.M.

TITLE OF THESIS: Reverse Engineering Computer Programs Under Canadian Copyright Law

Authorization is hereby given to McGill University to make this thesis available to readers in the McGill University Library or other library, either in its present form or in reproduction. The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise be reproduced without the author's written permission. The authorization is to have effect on the date given below unless a deferral of one year is requested by the author on submitting the thesis.

Signature of Author: Sunny Handa

Date: Aug. 30/94

Permanent Address of Author:
8 Holmdale Rd.
Hampstead, Quebec
CANADA
H3X 1Y1

Abstract

The field of copyright law has been especially active in recent times as a result of its application to computer programs. Copyright law, not originally designed to protect such works, has had to adapt to suit the special nature of computer programs. This paper addresses the applicability of copyright law to the reverse engineering of computer programs. Reverse engineering is a method by which programmers may uncover the ideas and processes used within an existing computer program, thereby allowing the construction of compatible computer programs. Reverse engineering may also be used to create works which are directly competitive with the original program, and may also be used to assist in the piracy of computer programs. The mere act of reverse engineering computer programs, regardless of its purpose, potentially infringes the copyright of the computer program in question, notwithstanding whether the results of the process are used in an infringing manner.

Recently both the European Union countries and the United States have accepted reverse engineering as an exception to copyright infringement. The European Union has opted for a legislative solution, whereas in the United States several courts have construed the fair use exception contained in that country's *Copyright Act* as allowing reverse engineering.

In this paper, it is argued that Canada must also adopt a reverse engineering exception to copyright infringement. It is claimed that the implementation of such an exception is justified through examination of the underlying policy goals of copyright law in the context of an economic framework. Reverse engineering fosters the creation of standards which, it is argued, increase societal wealth. The existence of a reverse engineering exception is consistent with the balance between the economic rights of individual authors and societal technological progress, which copyright seeks to maintain. It is demonstrated that copyright exists as the only form of applicable intellectual property protection which can broadly limit the disclosure of concepts underlying computer programs.

It is suggested that an effective exception should be statutorily based. It is felt that the existing fair dealing exception contained in the Canadian *Copyright Act* is juridically under-developed and too uncertain to provide an effective solution to the reverse engineering problem. A legislative solution would send a clear message to the software industry as well as to the courts, and could prohibit contracting out of the *Copyright Act* which would potentially be allowed were a judicial solution sought. It is further suggested that the statutory exception should broadly allow the process of reverse engineering as opposed to limiting it to cases where compatibility is sought. Narrowing the exception creates conceptual difficulties in applying limits to reverse engineering. Allowing a broad exception would avoid these difficulties while continuing to provide copyright holders with protection if, after the reverse engineering process is concluded, their protectable expression is used within another's software product.

Avant-Propos

La loi sur les droits d'auteurs a été l'objet de modifications récemment concernant son application aux logiciels d'ordinateur. Cette loi, qui n'était pas conçue à l'origine pour protéger de tels ouvrages, a dû être adaptée à la nature particulière des logiciels. Ce mémoire vise à analyser l'application des droits d'auteurs au processus d'analyse et de recomposition des logiciels, aussi appelé processus de décompilation. Ce terme désigne une méthode utilisée par les programmeurs pour extraire les idées et le processus utilisés dans un logiciel existant afin de produire de nouveaux programmes compatibles avec celui-ci. La décompilation pourrait, également, contribuer à la création de logiciels entrant directement en concurrence avec l'original ou même au piratage de logiciels. Le simple acte de décompiler des programmes informatiques, sans égard au but visé, viole potentiellement les droits d'auteurs sur un logiciel peu importe que les résultats soient ou non utilisés illégalement.

Récemment, tant les pays du marché européen que les États-Unis ont exclu la décompilation des infractions à la loi des droits d'auteurs. Le marché européen a opté pour une solution législative alors que, aux États-Unis, diverses cours ont interprété l'"exception de l'utilisation équitable" ("fair use exception") de l'acte des droits d'auteurs de ce pays comme permettant le processus de décompilation.

Dans ce mémoire, il est soutenu que le Canada doit aussi adopter une solution qui exclut le processus de décompilation des droits d'auteurs. Il est argué que l'implantation d'une telle exception est justifiée par l'analyse des principaux objectifs visés par la loi concernant les droits d'auteurs dans le contexte d'un encadrement économique. Le processus de décompilation aide à créer des normes qui, selon l'argumentation soumise, contribuent à l'enrichissement de la société. L'exemption de ce processus constitue une démarche logique dans le cadre d'un équilibre entre les droits économiques des auteurs individuels et le progrès technologique de la société que la loi sur les droits d'auteurs cherche à soutenir.

Il est suggéré qu'une exception effective devrait être attestée par une loi. Il est soutenu que l'exemption pour l'utilisation équitable contenue dans la loi canadienne sur les droits d'auteurs est, juridiquement parlant, insuffisante et trop vague pour offrir une solution efficace au problème du processus de décompilation. Une solution législative transmettrait un message clair à l'industrie des logiciels ainsi qu'aux cours de justice et prohiberait l'option d'exclusion volontaire ("contracting out") de la loi sur les droits d'auteurs. Ce exclusion serait possiblement accordée si une solution judiciaire était privilégiée. En plus, il est suggéré que l'exemption statutaire englobe tout le processus de décompilation plutôt que de le limiter aux cas où une compatibilité est recherchée. Contraindre l'exemption crée des difficultés conceptuelles dans l'application des limites au processus de décompilation. Accorder une exemption plus large les évite tout en assurant un recours aux détenteurs des droits d'auteurs si l'oeuvre protégée a été utilisée dans une autre production informatique suite au processus de décompilation.

Originality is nothing but judicious imitation. The most original writers borrowed one from another. The instruction we find in books is like fire. We fetch it from our neighbor's, kindle it at home, communicate it to others, and it becomes the property of all.
--Voltaire

Nothing can with greater propriety be called a man's property than the fruit of his brains. The property in any article or substance accruing to him by reason of his own mechanical labour is never denied him: the labour of his mind is no less arduous and consequently no less worthy of the protection of the law.
--Copinger and Skone James on Copyright

Laws that do not embody public opinion can never be enforced.
--Elbert Hubbard

If we desire respect for the law, we must first make the law respectable.
--Louis D. Brandeis

Table of Contents

Chapter I. Introduction

The field of computer law has grown in leaps and bounds in recent years. In fact, the very use of the title "computer law" as denoting a separate area of law continues to be regarded with some skepticism by traditionalists. It is undeniable, however, that a number of law firms now house departments or groups that specialize in this fast growing field and that lawyers hold themselves out as computer lawyers. What distinguishes computer law from accepted traditional legal categories such as copyright or, even more broadly, intellectual property law is that computer law is inter-disciplinary, drawing from many existing and distinct areas of law including, but not limited to, copyright, patents, trade-secrets, semi-conductor chip, contract law, criminal law, and tort law.

In terms of civil and criminal protection against illicit copying of computer programs, copyright law has clearly evolved as the standard form of protection throughout the world. International copyright conventions such as the Berne Convention[1] and the Universal Copyright Convention[2] now explicitly refer to computer programs as protectable works. The field of copyright law, often thought of as settled and slow

---

1 *Revised Berne Convention*, (1886 as amended to 1928) as reprinted in *Copyright Act*, R.S.C. 1985, c.C-42 as amended, Sch. II. Canada is currently a signatory to the 1928 Rome revision of the *Berne Convention*. Canada is not a party to the 1948 Brussels revision, the 1967 Stockholm revision, or the 1971 Paris revision of the Convention. However, the North American Free Trade Agreement, to which Canada is a party and which came into force on January 1, 1994, requires that the contracting member states accede to the [latest] 1971 Paris revision to the *Berne Convention*.

2 Section 2.1 of the *Copyright Act*, R.S.C. 1985, c.C-42 as amended [hereinafter the *Copyright Act*], extends copyright protection to nationals of countries who have adhered to the *Universal Copyright Convention*, adopted on September 6, 1956 in Geneva, or to that Convention as revised in Paris on July 24, 1971.

moving, has received an infusion of activity within the past decade as a result of its application to computer programs. Computer programs, because of their very nature, while broadly fitting the underlying framework of copyright law do not lend themselves perfectly to many of the idiosyncratic jurisprudential concepts of copyright law that have developed in respect of traditional works over the past century. Accordingly, copyright law as it applies to computers is rapidly developing its own wealth of jurisprudence.

One of the most topical and difficult decisions facing legislators and jurists in this area is the question of the permissibility of reverse engineering of computer programs under copyright law. Reverse engineering, or decompilation as it is sometimes referred to, involves taking a finished product and working backwards in order to gain a better understanding of how the product was produced. The question of whether works which are protected by copyright may be legally reverse engineered without the copyright holder's consent remains unchallenged under Canadian law. Indeed, only a few jurisdictions have yet had occasion to deal with the issue. Most notably, the European Union, formerly the European Economic Community, in its European Software Directive of May 14, 1991,[3] took a bold step forward in declaring that one is free to decompile computer programs for the purposes of achieving interoperability with other computer programs. More recently, several U.S. courts have had to deal with whether and under what circumstances reverse engineering of computer programs would be permitted under

---

3 *European Council Directive of May 14, 1991 on the Legal Protection of Computer Programs* (91/250/EEC).

U.S. copyright laws.[4] The issue of reverse engineering of copyrighted works has not, to-date, been dealt with in Canada either through explicit legislation or by the courts. A broad reading of the Canadian *Copyright Act* suggests that *prima facie* the reverse engineering of a copyrighted work is prohibited without the copyright holder's consent.

The U.S. copyright legislation, however, while appearing very similar to the Canadian legislation, has been interpreted by U.S. courts as allowing the reverse engineering of computer programs in certain circumstances. Although Canadian courts are not bound by their American counterparts, there is much that is borrowed from American law in the computer law field.[5] There are also strong public policy arguments to

---

4 *Sega Enterprises Ltd. v. Accolade Ltd.*, 24 U.S.P.Q.2d 1561 (9th Cir. 1992); and *Atari Games Corp. v. Nintendo of America Inc.*, 975 F.2d 832 (Fed. Cir. 1992). For cases involving the reverse engineering of data tables see: *E.F. Johnson Co. v. Uniden Corp. of America*, 623 F.Supp. 1485 (D.C. Minn. 1985); and *Autodesk Inc. V. Dyason* (1990), 18 I.P.R. 109 (Aust. Fed. Ct.), reversed (1992), A.I.P.C. 90,855, 22 I.P.R. 162 (Aust. H.C.).

5 In *Compo Co. Ltd. v. Blue Crest Music Inc.* (1979) 45 C.P.R. (2d) 1 (S.C.C.), Estey, J. stated,

> The United States *Copyright Act*, both in its present and earlier forms, has, of course, many similarities to the Canadian Act, as well as to the pre-existing *Imperial Copyright Act*. However, United States' court decisions even where the factual situations are similar, must be scrutinized very carefully because of fundamental differences in copyright concepts which have been adopted in the legislation of that country. ... That is not to say that we may not find some assistance in examining the experience in the United States ... ".(At p. 8)

O'Leary J. further recognized the value of American jurisprudence in *Delrina Corp. v. Triolet Systems Inc.* (1993), 47 C.P.R. (3d) 1 , Court file no. 12515/86 (Ont. Ct. Gen. Div.), stating,

> The United States *Copyright Act* differs somewhat from Canadian *Copyright Act*, but nevertheless American copyright decisions were heavily relied on by both the plaintiff and defendants in this case and are of great assistance on the issues before me. (At p. 28).

be made in favour of allowing the reverse engineering of computer programs. Reverse engineering allows for the creation of interoperable, or compatible, computer programs in cases where the copyright holder will not release the program's technical specifications. The balance of public benefit versus the protection of an individual's right to claim rewards associated with a works' distribution and use is at the core of intellectual property rights protection. The societal benefits to be gained by allowing reverse engineering must be weighed against the potential risks to the copyright holder associated with allowing reverse engineering.

In this paper, it will be suggested that a broad right of reverse engineering with respect to computer programs should be permitted under Canadian law. This claim will be based on a cost/benefit, or economic, analysis of the law and the outcomes associated with various proposed reverse engineering scenarios. Justification of the use of law and economics as a theoretical basis of support for allowing reverse engineering, and as an explanation of intellectual property protections more generally, will be discussed in Chapter V. This economic approach will be contrasted with other theoretical justifications for intellectual property law such as a means of providing cultural protection or guaranteeing that moral claims to a work's authorship or invention are properly attributed.

Chapter II. Concepts

Before proceeding with any further discussion of the reverse engineering of computer programs it is necessary to define concepts such as reverse engineering, computer programs and other related concepts both for purposes of clarity and consistency. This Chapter is not meant to provide a comprehensive review of computer technology but, rather, is meant to provide an uninitiated reader with a cursory overview of the technological concepts involved in the reverse engineering debate.

A. Computer Programs

Section 2 of the Canadian *Copyright Act* defines a "computer program" as "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." This definition is virtually identical to that set out in Section 101 of the U.S. *Copyright Act*[6] which defines "computer program" as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." Section 342.1(2) of the Canadian *Criminal Code*[7] defines a computer program as "data representing instructions of statements that, when executed in a computer system causes the computer system to perform a function".

More detailed definitions can be found scattered throughout the jurisprudence dealing with the copyrightability of computer programs. Ferris J., in *John*

---

[6] U.S. *Copyright Act*, 17 U.S.C. § 101.

Richardson Computer's Ltd. V. Flanders and Chemtec Ltd.,[8] held that a computer program is "a list of instructions or routines or actions set out in a logical order and designed to solve a particular problem. A series of such instructions may be combined together in order to solve a more complex problem, or a number of problems. Such a series ... may equally be described as a single program."[9] All of these definitions commonly hold that a computer program consists of an arrangement of instructions that is used in a computer to solve a particular problem. Simple data that does not in itself instruct a computer to perform calculations towards a given end, does not qualify as a computer program.

Computer programs can broadly be categorized into two types: operating system programs and application programs. Reed J. articulated a particularly good definition of each in *Apple Computer Inc. V. Mackintosh Computers Ltd.* where Her Ladyship remarked,

> Application programs are designed for a specific task, such as the playing of a video game, preparation of a tax return, or the writing of a text. Operating system programs are designed primarily to facilitate the operation of application programs and perform tasks common to any application program. Without them each application program would need to duplication their functions.[10]

---

7 *Criminal Code*, R.S.C. 1985, c. C-46, as amended.

8 *John Richardson Computers v. Flanders and Chemtec* (U.K. High Court, February 19, 1993).

9 *Ibid*, at p. 1.

10 *Apple Computer Inc. V. Mackintosh Computers Ltd.* (1986), 10 C.P.R. (3d) 1 (F.C.T.D.); varied (1987), 44 D.L.R. (4th) 74 (Fed. C.A.), aff'd [1990] (S.C.C.), per Reed J. (F.C.T.D.) at p. 11.

B. Computer Languages

Also intrinsic to understanding the issues surrounding reverse engineering is a minimal knowledge of how a computer program is built, compiled and executed as well as an understanding of the related jargon. Computer programs are written in computer languages which vary in their degree of resemblance to "ordinary mathematics and English (or other common languages)".[11] A higher level language is said to be closer to "common languages" in its vocabulary than a lower level language. The level, also referred to as the generation, of the language depends "upon the ease with which it can be read" by human beings.[12] In order for a computer to process the instructions of any given language, the instructions must first be compiled, or translated, into a language or notation that the computer's processor can understand. This latter notation is known as the lowest, or first, level language.

Fourth generation languages, or "4GLs", consist of database languages used primarily by end users rather than professional programmers. 4GL commands often use entire English words and may resemble the following: "DO UNTIL NUMBER_OF_CUSTOMERS IS 10".

Slightly more cryptic are third generation languages which consist of C, PASCAL, COBOL, BASIC, FORTRAN, and other similar languages. These languages

---

11 *Ibid*, at p. 7.
12 *Ibid.*

are commonly used by programmers and consist of some English words combined with a greater level of numerically represented computer logic. A 3GL instruction in BASIC similar to the 4GL example above might be: "FOR CUSTOMER = 1 TO 10; [insert other instructions]; NEXT CUSTOMER".

Second generation languages, referred to as intermediate level, as opposed to high level, languages consist of ASSEMBLER languages. ASSEMBLER language instructions are comprised of mnemonic instructions combined with memory addresses, usually denoted in hexadecimal notation (number system with a numerical base of 16), and are used by programmers for performing specialized tasks that require extremely efficient programming. Practically speaking, ASSEMBLER is the lowest level language used by programmers with few exceptions. A typical ASSEMBLER instruction might be "JMP" followed by a memory address that instructs the computer to branch, or jump, to the instruction indicated by the address.

The lowest level language is known as MACHINE LANGUAGE, and is often referred to as "object code". MACHINE LANGUAGE is often represented in either binary (number system with a numerical base of 2) or hexadecimal notation. MACHINE LANGUAGE in binary form, made up exclusively of "1"s and "0"s, can be understood directly by a computer's central processing unit without need of any further compilation. The quick explanation for this is that the "1"s and "0"s, known as bits,

represent on and off states which when converted to on and off voltage states trigger countless switches or gates contained in a computer's processor.[13] The triggering of these gates creates a domino effect with other gates producing an effect that is translated through the computer's hardware into a real world event (some form of output or calculation).

In sum, the term source code refers to the written form of program that the user physically produces in a given computer language.[14] Programmers today have a wealth of higher languages from which to choose including PASCAL, C, FORTRAN, BASIC, ASSEMBLER, and so forth. Once the program's source code has been written, the programmer will "compile" the source code into machine readable object code using another computer program known as a compiler to perform the conversion. Object code

---

13 People often confuse the terms "bit" and "byte". While the former represents binary information (0's and 1's) the latter is used to represent alphanumeric characters. Eight bits ("binary digit") make up one byte. As a consequence, there are two hundred and fifty six possible characters which may be represented by a byte at any given time (2^8 bits = 256). The term "at any given time" refers to the fact that different character sets may be used to represent each of the 256 possible characters. A commonly accepted character set is the American Standard Code for Information Interchange ("ASCII") which employs a seven bit scheme. Consequently there are only one-hundred and twenty-eight characters (2^7 bits = 128 characters). There are many variations of ASCII which use the eighth bit to expand the character set to two-hundred and fifty six characters.

14 Source code refers to a set of

> [c]omputer instructions that are written in a structured programming language that is human readable. The opposite of "object code". The instructions required to define the processing steps required expressed in a format that the human programmers can more easily work with. This format of code is not readily understandable by the computer but can be interpreted more easily by the programmer. The notation used to express the instructions is referred to as a computer language.

Delrina Corp. v. Triolet Systems Inc. (1993), 47 C.P.R. (3d) 1, Court file no. 12515/86 (Ont. Ct. Gen. Div.), at p. 53.

is generally in binary form - a language made up exclusively of "1"s and "0"s - and is directly useable by the computer. A programmer may, of course, write his program directly in binary form, but this is not often done due to obvious conceptual difficulties.[15]

C. *Computer Memory*

Another important concept central to the use of computers is memory. The Canadian *Copyright Act*'s definition of computer program set out above requires that a set of instructions be "expressed, fixed, embodied or stored" in order to qualify as a computer program. In the United States the legislation defines this fixation as existing where a tangible mode of expression is embodied in a form which is sufficiently permanent and stable so that it may be "perceived, reproduced, or otherwise communicated for a period of more than transitory duration, either directly or with the aid of a machine or device."[16] Traditionally, the fixation of copyrighted literary works was done on paper.[17] This type of storage, while also remaining valid for computer programs, is not the only type of fixation possible. Computer programs that are stored in a computer's memory device also qualify as being stored for the purposes of the *Copyright Act*.[18]

---

15 "Although it is possible for engineers to write software in machine language, the process is extraordinarily tedious, and is virtually never done", Gary R. Ignatin, "Let the Hackers Hack: Allowing the Reverse Engineering of Copyrighted Computer Programs to Achieve Compatibility", 140 University of Pennsylvania Law Review 1999, at p. 2001. See also, Dennis S. Karjala, "Copyright, Computer Software, and the New Protectionism", (1987) 28 Jurimetrics J. 33, at p. 37.

16 17 U.S.C. § 102.

17 For a detailed discussion of "fixation" see *infra*, Chapter III.A.1.d. Fixation, at p. 35.

18 In *Apple Computer Inc. v. Macintosh Computer Ltd.* (1986), 28 D.L.R. (4th) 178 (Fed. T.D.), varied (1987), 44 D.L.R. (4th) 74 (Fed. C.A.), aff'd [1990] (S.C.C.), the Court held that object code which was stored on a silicon micro-chip was a reproduction in a material form of copyrightable source code and was therefore protectable as a "computer program" under the

Today's computers use a variety of memory devices to store their programs. For the purposes of convenience, these devices can be separated into three categories: internal, external and archival.[19] [20] Internal memory is simply those memory devices, built in to the structure of the computer, that are necessary for the computer to operate at its most basic level. External memory are devices that provide additional, more permanent, storage at relatively cheap cost. These devices are not essential for the basic operation of the computer but are nonetheless required for practical purposes. Archival memory, consisting of devices such as tape-backup machines, does not impact on the reverse engineering debate in any meaningful way and will not be discussed further.

The most commonly used external memory devices are disks, made up of both hard disks and floppy diskettes. Disks are magnetic media that hold vast amounts of data relatively inexpensively.[21] Disk drives, the device that interfaces a disk with a

---

Copyright Act. According to the Court, it was irrelevant that the object code was not necessarily in human readable form.

19 Raymond R. Panko, End User Computing, John Wiley & Sons (New York, 1988), at p. 315.

20 A simple, but useful, definition of "memory" can be found in *Delrina Corp. v. Triolet Systems Inc.*, *supra*, note 14, at p. 51:

> [memory is defined as a]n area of the computer's circuitry that holds applications and any data generated with those applications. Information held in Random Access Memory (RAM) is erased whenever the computer is turned off. Information held in Read Only Memory (ROM) is retained even when the computer is off. Memory usually refers to the high speed semiconductor storage within a computer that is used to temporarily store data while it is being processed or examined. The term "memory" is also generically extended to refer to data that is stored externally on disks and tapes.

21 Magnetic tape was the storage medium of choice prior to the advent of the disk/ette. Disk/ettes allow the user to access the media in a random, as opposed to a sequential, fashion unlike magnetic tape.

computer, are capable of reading and writing information to disks. A newer technology that is gaining mass popularity is the CD-ROM (compact disc read-only-memory), another external device. The CD-ROM operates using laser technology and is capable of storing even greater amount of information than disks, also at inexpensive prices.[22] Currently the mass market CD-ROM's are sold read-only, with the information encoded on the disc at the time of purchase, and the user cannot store information on the CD-ROM.

Another type of memory device (internal) is a ROM-chip (read-only-memory microchip). Like the CD-ROM, ROM-chips are encoded at the factory and cannot be written to once encoded. ROM-chips are microchips which are silicon based computer chips that store information using a system of microscopic gates that route electrical impulses to their intended destination based on the programming in the chip. Many variants of ROM-chips, such as EEPROM (electronically-erasable programmable read-only-memory) chips, also exist which support different characteristics such as the ability to store non-volatile[23] data on the chip. ROM chips often contain a host of "service

---

Disk/ettes, like tapes, vary in their size and storage capacity. The capacity of disk/ettes has steadily improved over the past two decades. The current diskette standard is 3 ½" in length and width with a storage capacity of 2 megabytes (one byte is equal to eight bits) of information, although the popular IBM PC standard on uses a capacity of 1.44 megabytes per disk. Disks (i.e. hard disks) now vary in storage capacity, which is now commonly measured in the hundreds of megabytes.

22 The size of a CD-ROM (and compact disc) is 120mm in diameter, or about the size of a cardboard thin doughnut, and is capable of storing b/w 550 and 600 megabytes of digital data, the equivalent of roughly 400 high density 3 ½" IBM standard computer diskettes. (*Supra*, note 19, at p. 255).

23 Volatility in the context of microchips refers to the characteristic whereby a continuous supply of power is required to maintain storage of the information in the chip. A non-volatile chip maintains its storage without a continuous supply of power.

programs" that interact with a computer's processor chip's limited instruction set,[24] in order to perform frequently requested tasks such as accepted keystrokes from the keyboard and checking the state of various add-on devices such as external memory devices.[25]

Next to disk storage, the most commonly known type of computer memory is probably RAM (random-access memory) chips. RAM, also known as volatile or dynamic memory, is a form of internal memory that stores information as long as electrical impulses are being fed through it. Once the power is cut-off, the RAM microchips loose all information that was stored in them. RAM is the functional memory that allows a computer to operate, as the programs which the computer processes, or parts thereof, must be stored in RAM during the operation of the computer (with the exception of ROM programs). If programs are otherwise stored on external memory devices, they must be copied into RAM memory in order to be executed.[26]

---

24 A computer's processor, often referred to as a central processing unit (CPU) or microprocessor chip, is the "brains" of the computer. The CPU (or CPUs in the case of computers with multiple processors) performs all of the calculations and computing tasks. CPUs contain a limited number of instructions that are activated through electrical impulses that enter the processing chip. The number of instructions will vary with the architecture of each CPU. The effectiveness of reduced instruction set computer (RISC) processors, which contain few simple instructions, and complex instruction set computer (CISC) processors, which contain many instructions at the processor level, have been hotly debated in the popular press with respect to the recent releases of Motorola's Power PC processor (RISC) and Intel's Pentium processor (CISC).

25 *Supra*, note 19, at p. 320.

26 *Ibid*, at p. 315 .

D. Operating Systems

In order for a computer to execute its programs, operate its devices, and interact with the user, an operating system is required. As mentioned, an operating system is a computer program that interfaces between the computer and application programs that the user wishes to run.[27] The purpose of the operating system is to set up and manage the computer system's environment and resources such as input/output devices (keyboard, screen, printers, etc.), memory usage, and low level interpretation of instructions that are sent from the application program to the computer's micro-processor. In effect, the operating system expands upon the limited functions contained in the ROM.[28] While understanding the technical operation of operating systems is not required for the purposes of this discussion, it is important to understand that operating systems are computer programs whose specifications are essential to computer application programmers who wish to write computer programs that operate on given computer systems.[29]

---

27 In *Delrina Corp. v. Triolet Systems Inc.*, *supra*, note 14, at p. 52, O'Leary J. defined an operating system as:

> A set of programs, usually supplied by the manufacturer of a computer system that manages the basic operation of the computer system including such things as saving and retrieving data, providing security between users and backing up the data for archival purposes. Common examples of these include MSDOS (IBM compatible PC's), MPE (HP3000), UNIX (various computers), VMS (Digital Vax).

28 *Ibid*, at p. 320.

29 Common examples of operating systems are: AT&T's UNIX, Microsoft's DOS, Apple's System 7, and Microsoft's Windows (which currently piggybacks on the DOS system).

Additionally, operating systems, in order to interact with the user, consist of a user interface which is defined as that part of a computer program "that interacts with the user, generally consisting of the layout of screens, sounds, command sequences, and so forth."[30] [31] Individual application programs also contain their own user interface, although many application program user interfaces will try to remain consistent with the operating system user interface in order to increase the ease of use of the program by providing the user with a familiar environment within which to navigate.[32] In effect, the operating system provides the programmer and the user with a standardized environment with which is used to interact with the computer's hardware.[33] The importance of creating standards is vital to the reverse engineering debate as will become apparent in the following chapters.

---

30 Sunny Handa, "Have Recent Copyright Decisions Unduly Suppressed the Emergence of Standardized User Interfaces" (Paper written at the Faculty of Law, University of Toronto, 1991) [unpublished], at p. 1.

31 Keeton J., in *Lotus Development Corporation v. Paperback Software International and Stephenson Software Ltd.*, 740 F.Supp. 37 (D. Mass. 1990), accepted that for the purposes of that case "user interface" included "such elements as the 'menus (and their structure and organization) the long prompts, the screens on which they appear, the function key assignments, [and] the macro commands and language'". (At p. 63).

32 For instance, the use of a mouse to move an arrow on the screen followed by the press of a mouse button which allows the user to complete operating system tasks, such as running an application program, will also be present in the application program within which the user will use the arrow and the press of a mouse button to complete application program tasks. Although the appearance of the arrow and handling of the mouse may be changed by the application program, this is impractical and is not often done.

33 Operating systems may be in the form of software (i.e. stored on disk), hardware (i.e. hard coded on a ROM chip) or a combination of both. In *Apple Computer Inc. V. Mackintosh Computers Ltd* the successful copyright claim of infringement concerned an operating system stored on a ROM chip. Microsoft's popular DOS operating system is stored in disk form, allowing the Microsoft to upgrade it easily. Apple's System Seven operating system is stored both on disk and on ROM chips.

E. Reverse Engineering

Reverse engineering, as the name suggests, is opposite to the process of constructing a computer program as described. Reverse engineering, also known as disassembly or decompilation, "involves going backwards from a finished product and determining how the product works".[34] Another definition holds that reverse engineering occurs where "one inspects or takes apart a new product ... by translating the unreadable object code of a program into source code that may be studied."[35] The terms "disassembly" and "decompilation" which are synonymous with "reverse engineering" are actually subsets thereof. Decompilation of a computer program occurs where one "convert[s] the machine code version [of the program] into a high level language",[36] whereas "[d]isassembly of a computer program is done by translating the machine or object code into humanly-readable assembly language".[37] The only difference between decompilation and disassembly is the product obtained at the end of the process. In the former case it involves converting the machine code into a high level language whereas in the case of disassembly the final product is in ASSEMBLER, an intermediate level, language. Some commentators argue that technically "[d]ecompilation is only possible if the source code was in a high level language and the precise version of that language is

---

34 *Sega Enterprises Ltd. V. Accolade Inc.*, 23 U.S.P.Q.2d 1440 (D.C. N.D. Ca. 1992), at p. 1441.

35 Gary R. Ignatin, "Let the Hackers Hack: Allowing the Reverse Engineering of Copyrighted Computer Programs to Achieve Compatibility", 140 University of Pennsylvania Law Review 1999, at p. 2010.

36 David I. Bainbridge, "Computer Programs and Copyright: More Exceptions to Infringement", (1993) 56 The Modern Law Review 591, at p. 593.

37 *E.F. Johnson Co. V. Uniden Corp. Of America*, 623 F.Supp. 1485 (D.C. Minn. 1985), at p. 1490.

known and is performed by using a computer program to carry out the conversion".[38] Although semantically this may be correct, it is possible to construct a program that can convert a machine language program into a high level language other than that used in programming the source code. Such a conversion would also loosely qualify as decompilation.

Practically speaking, most reverse engineering of computer programs is of the disassembly variety as the computer programs that are used in performing the disassembly are easier and more flexible to create than are decompilers. Furthermore, software engineers and computer programmers involved in reverse engineering are generally quite comfortable in an ASSEMBLER language environment and do not need to visualize the program in a higher level language.

1. Intermediate Copying

The copying of computer programs, as it relates to the reverse engineering process, can occur in several ways. The first instance of copying that results from reverse engineering occurs during the deconstruction process. Whether the reverse engineering process is conducted through a manual inspection of the program code which is reassembled on paper, or through the more common method of disassembly, the process invariably results in copying.[39] This copying does not involve creating a completely

---

38 *Supra*, note 36, at p. 593.

39 Copying, for the purposes of copyright, is subject to the copies being "fixed" in some form. See *infra*, note 104, and accompanying text, for a discussion of fixation under copyright law.

verbatim copy of the original code, but rather, consists of translating the original program code several times, each moving towards a assembler, or higher level language translation of the original object code.[40] Whether intermediate copies produce during the disassembly process violate copyright rules is not altogether clear and will be more fully discussed below.[41]

2. Reverse Engineering and Piracy

The most widely known form of illicit copying, piracy, concerns the direct reproduction of a computer program, usually by a user, without the author's consent. Because computer programs are stored digitally (as 1s and Os), flawless reproductions can be made at little cost to the copier.[42] Because of this fact, computer programs, and more recently digitally stored audio recordings, have increased the need for intellectual property protection as the economic incentives that result in purchases of the original

---

40 A disassembler makes several "passes" over the original code, gradually building towards a final translation that is in assembler language.

41 See *infra*, note 108 for a discussion of intermediate copying that remains stored only in RAM. Where the intermediate copies are stored in a more permanent manner, it is more likely that they will violate copyright laws. See *infra*, Chapter III. B. Is Reverse Engineering An Infringement Of Copyright Law?", at p. 50. Additionally, see section 3(1)(a) of the *Copyright Act* which prohibits unauthorized translations of protected works; and see *infra*, note 257, and accompanying text for the decision of the U.S. Court of Appeals (9th Circuit) with respect to intermediate copying.

42 In its *Final Report* to Congress concerning the copyrightability of computer programs, the U.S. National Commission of New Technological Uses of Copyrighted Works ("CONTU") stated that,

> The cost of developing computer programs is far greater than the cost of their duplication. Consequently, computer programs ... are likely to be disseminated only if ... the creator can spread its costs over multiple copies of the work with some form of protection against unauthorized duplication of the work ... (CONTU, *Final Report* (1978), at 20-21.

product have diminished.[43] Typically, with this form of direct copying, a user will make an unaltered copy of an original[44] computer program, or of an existing copy thereof, and will use the copy in place of purchasing the original computer program.[45] Generally these copies can easily be produced using basic operating system commands such as "copy" or "diskcopy". In order to deter this copying, some computer program manufacturers have attempted to use various copy protection schemes in order to deter this practice.

Copy protection schemes vary in their functioning and because of their very purpose, no standards can exist. However, these schemes can largely be boiled down to three basic types:[46] (1) the program is stored in such a way that copying programs cannot copy all the necessary parts; (2) the program prompts the user for a code or other piece of information that can only be found in the original packaging; or (3) the program will come with a hardware device that attaches to the computer and will send the program signals or information which the program will seek prior to functioning. The protection type first mentioned is often defeated by third party developed copying programs that copy the required parts. With respect to the latter two schemes, invariably

---

43 William Landes and Richard Posner, "An Economic Analysis of Copyright Law", (1989) 189 Journal of Legal Studies 325, at p.327.

44 By "original" we mean a copy of the work produced with the authority of the copyright holder, as opposed to a "copy" which is produced in the absence of such permission.

45 "Persons who have not paid for a software copy cannot be excluded from using a program, and use of a program copy by one person does not necessarily diminish the supply of copies available for use by others", David A. Rice, "Public Goods, Private Contract and Public Policy: Federal Preemption of Software Licence Prohibitions Against Reverse Engineering, (1987) 53 Univ. of Pitt. L.R. 543, at p. 545.

46 Other, more eclectic, forms of copy protection exist however a comprehensive review of these schemes is beyond the scope of this paper.

program modifications, known as "cracks", designed to defeat the protection will appear soon after a program's public release.[47]

Program cracking in order to defeat a copy protection scheme often involves some disassembly of the protected computer program. Cracking programs, or enhanced disassemblers with specialized features to assist a cracker, are also generally freely available as either shareware or as freeware.[48] Recently, the reverse engineering debate was brought in front of the U.S. courts with respect to video game cartridges which contained computer programs protected with a program check (akin to protection type (3) set out above).[49]

3. Using the Results Obtained Through Reverse Engineering

The other form of illicit copying occurs where the copier, usually a programmer, alters or uses parts of the original program in his/her own work. The amount of modification varies greatly in this range and may or may not be substantial

---

47 A "crack" is often distributed in either printed form (as a set of instructions on how to modify the program to defeat the protection) or as a "patch". A patch is a small computer program that applies itself to the protected program and replaces the required code with a replacement that defeats the protection. An example of a simple crack is a set of instructions that tells the program to skip over the code that executes the protection checks. "Cracks" are commonly distributed on various computer bulletin boards and are easily available on the Internet – the global computer network.

48 Shareware refers to computer programs which may be used for a trial period without infringing copyright, after which a licence fee is payable to the copyright holder for continued use. Shareware programs are also freely distributable in their unaltered state to other users (hence the "share" in shareware) who may try them out for the trial period without payment. Freeware refers to computer programs where the copyright holder waives his/her rights to any economic return for its use. Waiving of economic returns does not mean a waiver of moral rights which would allow users to modify and alter the original work. Any such waiver is independent of the free/shareware designation.

49 *Sega Enterprises Ltd. v. Accolade Ltd.*, 24 U.S.P.Q.2d 1561 (9th Cir. 1992); and *Atari Games Corp. v. Nintendo of America Inc.*, 975 F.2d 832 (Fed. Cir. 1992).

enough to violate copyright.[50] Copying a work in this manner may not require actual direct copying of the original program code (known as literal copying).[51] "Non-literal" elements of a computer program are "those aspects that are not reduced to written code"[52] and include "components such as general flow charts as well as the more specific organization of intermodular relationships, parameter lists and macros".[53] Screen displays also fall within the definition of non-literal elements.[54] Copying these non-literal elements may also be an infringement of copyright and may be accomplished without dissecting the program as previously discussed. For example, copying the layout of a screen may simply involve a visual examination of the original program and replicating it using entirely new programming. Similarly, copying the order of the keystrokes used in the operation of a computer program, known as command sequences, may not involve actual literal copying of the original code. The issue of non-literal copying of computer programs has been highly topical in recent years, and remains far from being resolved.[55]

---

50 Copyright only protects expressions and not the ideas that underlie them. Please refer to Chapter III for a discussion of copyright principles.

51 *Supra*, note 45, at note 79.

52 *Computer Associates International, Inc. V. Altai, Inc.*, 23 USPQ2d 1241 (2nd Cir. 1992), at p.1244.

53 *Ibid*, at p. 1249.

54 Screen displays are protected as parts of a computer program "except in the case of programs whose very purpose is to produce screen displays for use in playing of games or for some artistic or like purpose", *Delrina Corp. v. Triolet Systems Inc.*, *supra*, note 14, at p. 32.

55 *Delrina Corp. v. Triolet Systems Inc.* (1993), 47 C.P.R. (3d) 1 (Ont. Ct. Gen. Div.); *Systèmes Informatisés Solartronix v. Cégep de Jonquière* (1988), 22 C.I.P.R. 101 (Que. Sup. Ct.); *Lotus Development Corporation v. Paperback Software International and Stephenson Software Ltd.*, 740 F.Supp. 37 (D. Mass. 1990); *Computer Associates International Inc. v. Altai, Inc.* (Second Circuit U.S. Court of Appeals, June, 1992); and *John Richardson Computers v. Flanders and Chemtec* (U.K. High Court, February 19, 1993), to name a but few decisions, have all recently tackled the issue of non-literal infringement. The jurisprudence in the U.K. and Canada now borrow from the U.S. decision in *Computer Associates International Inc. v. Altai, Inc.* wherein the Second Circuit U.S. Court of Appeals developed the "abstraction - filtration - comparison" test used to determine whether specific non-literal elements, in that case the look and feel of the

Generally, reverse engineering will not involve non-literal copying of the visual type described but will instead consist of a deconstruction of the original program's code expression in a move towards uncovering its underlying ideas. Whether the reverse engineering will be sufficient to uncover these ideas, or will simply stop at the point of uncovering expression will depend on each individual case. This exercise will be followed by a working forward, using the results of the reverse engineering, in the construction of a different program. The degree to which the re-programming will involve copying of the original program code will also vary greatly. Within this range of copying will fall some forms of non-literal copying, such as reproducing the layout of the programs subroutines (structure), as well as literal copying, such as copying parts of the original program's code either directly or through a translation into another language.

The results obtained by reverse engineering a program can be used for a number of purposes, including: the programming of cracks to defeat a program's copy protection, the creation of a similar program or of a program that uses the same routines as the original program in order to save time and expense (avoids "re-inventing the wheel"), academic study of the program's underlying ideas and the techniques used in their expression, or the creation of compatible, or interoperable programs. While the former two goals do not usually garner much support as they involve an element of

---

user interface, constitute protectable expression or whether they more properly fall within the domain of ideas and are, as a result, not protectable under copyright law.

thievery, the latter two goals are often supported as acceptable justifications of reverse engineering.[56]

4. Using Reverse Engineering in the Creation of Compatible Programs

Although no statutory definition of "interoperable" or "compatible" program currently exists, the concepts are generally simple to define. Compatibility, also known as interoperability, is a measure of the degree to which one program will function in conjunction with another program. In order to create a compatible program, "a programmer must have a complete specification of the other program's 'interface' - a precise description of how the program receives, stores and/or outputs information".[57] Traditionally, compatible programs were written by the same company as their programmers had access to the necessary specifications. An example of compatible programs are the popular Word Perfect 5.1™ and Draw Perfect™ programs. The former is a word processor whereas the latter is a drawing program. Word Perfect™ users may use pictures created with Draw Perfect™ within their word processed documents. The pictures appear within the Word Perfect™ document as pictures and may be manipulated to a limited degree using various keystroke commands.

---

[56] *Supra*, note 35, at p. 2022. Allowing reverse engineering for the creation of interoperable programs is recognized as the sole justification in Article 6 of the E.U.'s Software Directive which permits reverse engineering. *Supra*, note 3.

[57] *Supra*, note 35, at p. 2023.

A more modern approach to the design of compatible programs is to create more robust operating systems, which handle a greater number functions, and provide users with consistent specifications for all types of data structures (objects) controlled by each operating system. Programmers writing computer programs for use with such operating systems are usually given access to the operating system's specifications by the operating system designers at minimal cost. Objects created by computer programs that follow the specification are then useable in other programs that also follow the specification. The programs are therefore made compatible without the programmers ever having seen or used each others' computer programs. The operating system acts as the standardizing link. Examples of such object oriented operating systems are Microsoft's Windows™[58] and Next's NextStep™ operating systems. The move towards creating standards through operating systems is a sound one. However, the creation of new types of objects not contemplated by the operating system designers may arise. In such cases, the standards become proprietary to the application's designers once again.[59]

---

[58] Under the popular Microsoft Windows™ operating system a system of standardized objects entitled object linking and embedding (OLE) is used. OLE allows users to share information created in other applications with the application they are using. For example, users of Microsoft's Word™ word processing program may either link or embed objects created with Microsoft's Excel™ spreadsheet program inside their Word™ document. A large part of the linkage and embedding of these objects is a function of the Windows™ operating system and not of any specific design made by either application's programmers.

[59] There is also the question of whether operating system designers, often part of the same company that designs various applications for use with that operating system, will release all of the specifications required to make the most effective use of the operating system environment. Clearly there is motive to withhold some of the technical information as to provide one's own company with a competitive advantage in the application program market. In 1993, the U.S. Department of Justice stepped up a Federal Trade Commission anti-trust investigation of Microsoft Corporation. Among the charges being investigated are claims by competitors in the application program market "that Microsoft unfairly uses secret features known as

5. Conclusion

As the law currently stands, in some cases the newly constructed program, or parts thereof, will infringe the original program's copyright and be considered a copy whereas in other cases the new program will not have a sufficient degree of copying to be considered a copy for the purposes of copyright law. This determination is made irrespective of whether the new program is compatible or whether the program is constructed to compete with the original as a similar product.[60] A more immediate question, however, is whether the actual reverse engineering of the computer program is an infringement in itself. Both of these issues will be discussed in the following Chapter.

---

[60] 'undocumented calls' and its advance knowledge of changes to MS-DOS and the related Windows software to place its competitors at a disadvantage. Ilene Knable Gotts, "Regulators Focusing on Antitrust Issues", The National Law Journal (January 24, 1994), at p. S12. *Prima* facie copyright law currently remains oblivious to any such distinction. A determination of copying under the *Copyright Act* concerns whether the code copied is "substantial" in quality and not quantity. However, if a newly constructed program is directly competing with an original work from which information was reverse engineered there is an increased likelihood that the parts used will be considered substantial as the quality of the parts used may appear to be of greater import than if they had been used in a compatible program which is more likely to appear different to a court both in appearance and in program structure. See also the decision in *SAS Institute Inc. v. S & H Computer Systems*, 605 F.Supp. 1816 (U.S.D.C., 1985), *infra*, note 111.

Chapter III. Intellectual Property Protections for Computer Programs

Computer programs are protected from illicit copying under a number of legal regimes. Copyright, patent, trade secret, and semi-conductor chip laws all provide intrinsic measures of protection against the unauthorized copying of a computer program. The term "intrinsic" is used to distinguish the protections granted by these regimes from those contractual provisions that can be fashioned by private parties through negotiating a private agreement that is customized to suit their own circumstances. The protections provided by each of the aforementioned regimes are enforceable at law notwithstanding the absence of specific contractual agreements between parties.

A. *The Law of Copyright*

Although the application of copyright law to computer programs is a relatively new convention, copyright has existed as a form of intellectual property protection, in one form or another, for roughly four centuries.[61] The original impetus for devising legal rules that eventually would evolve into what we know as copyright law was, ironically, borne of the Crown's desire to censor subversive material.[62] By the reign of Henry VII only the King's Printers were provided the right to copy printed works by Royal prerogative.[63] This right to copy, eventually known as "copyright", evolved into an

---

61 Copyright law, also referred to as Anglo-American copyright law or the common law of copyright, is to be differentiated from *droit d'auteur*, or continental, regimes which protect similar works but are primarily used in civilian jurisdictions. Copyright, as discussed in this paper, refers to those regimes which were borne out of a common Imperial ancestry.

62 Edward Earle, "The Effect of Romanticism on the 19th Century Development of Copyright Law", (1991) 6 I.P.J. 269 (1991), at p. 271.

63 *Ibid.*

economic right which was converted into lots by rights holders which could then be transferred or traded as a commodity. These copy-rights "might be exchanged, assigned, bequeathed, or further subdivided, just like other intangible rights".[64] Contemporaneously with the commodification of copyrights, was the emergence of a common law of copyright.

The vision of copyright as a grant of a property-type right began to emerge in the mid-17th century, "when Parliament abolished the Star Chamber ... [and] was forced to replace the Chamber's decrees with controls of its own".[65] Over the following fifty years, Parliament was faced with a flood of "subversive" material and attempted to stem the flow by passing the *Licensing Act of 1662*[66] which imposed a "good-Christian" requirement on published works. In 1709, Parliament passed the *Statute of Anne*,[67] a precursor to modern day Anglo-American copyright legislation. The *Statute of Anne* recognized the rights of authors of both published and unpublished works by granting them a time-limited exclusive transferable printing right. Persons already owning transferred rights at the time of the statute's enactment were declared to be owners of the right. The dual protections afforded by the common law and the *Statute of Anne* came

---

64 Hugh Amory, "'*De facto* Copyright'? Fielding's *Works* in Partnership, 1769-1821", (1984) 17 Eighteenth Century Studies 449, at p. 453.

65 *Supra*, note 62, at p. 273.

66 *Licensing Act of 1662*, (U.K.), 14 Cha. 2, c.33.

67 *Statute of Anne*, (U.K.), 8 Anne, c. 19.

into conflict[68] and were eventually merged into only the statutory right by the House of Lords in *Donaldson v. Becket*.[69]

1. The Copyright Act

Following the abolition of the common law of copyright, the U.K. Parliament passed a series of *Copyright Acts*[70] which eventually fathered passage of a Canadian *Copyright Act*[71] in 1921 which came into force on January 1, 1924.[72] The Canadian *Copyright Act* has continued in the tradition of its Imperial forefathers and explicitly states that no copyright or similar right shall exist in Canada other than under the Copyright Act.[73] [74]

---

68 The Court in *Millar v. Taylor*, (1769) 98 E.R. 201, 4 Burr. 2303 (K.B.), allowed this duality continue, stating that the *Statute of Anne* merely granted the common law protections Royal Assent but did not abolish them. Accordingly the Court in *Millar v. Taylor* re-affirmed the common law principle of a perpetual copyright. Mark Rose, "The Author as Proprietor: *Donaldson v. Becket* and the Genealogy of Modern Authorship", (1988) 23 Representations 51.

69 *Donaldson v. Becket*, (1774) 1 E.R. 837; 4 Burr. 2408.

70 *Copyright Act*, (1814) (U.K.), 54 Geo. 3, c. 156; *Copyright Act*, (1842) (U.K.), 5 & 6 Vict., c. 45; and *Copyright Act*, (1911) (U.K.), 1 & 2 Geo. 5, c. 46.

71 *Copyright Act*, C. 24; R.S.C. 1927, c. 32; R.S.C. 1952, c. 55.

72 The passage of the Canadian *Copyright Act* abrogated all the *Copyright Acts* of the Imperial Parliament that had formerly applied to the Dominion of Canada, as it then was.

73 *Copyright Act* (1985), s. 63. Prior to passage of the 1921 *Act*, there had existed, in Canada, a common law copyright. This right was substituted by a statutory right under section 42 of the 1921 *Act*.

74 In *Compo v. Blue Crest Music*, [1980] 1 S.C.R. 357, Estey J. stated,

> copyright is neither tort law nor property law in classification, but is statutory law. It neither cuts across existing rights in property of conduct nor falls between rights and obligations heretofore existing in the common law. Copyright legislation simply creates rights and obligations upon their terms and in the circumstances set out in the statute. This creature of statute has been known to the law of England at least since the days of Queen Anne when the first copyright statute was passed. It does not assist the interpretive analysis to import tort concepts. The legislation speaks for itself and the actions of the appellant must be measured according to the terms of the statute. (At p. 372).

Today's copyright laws are no longer limited to the protection of published and unpublished manuscripts. Copyright currently protects a host of works including dramatic, musical and artistic works as well as a substantially broadened category of literary works. These broad categories can be further expanded to expose an even greater number of protectable works such as: tables, compilations, photographs, engravings, sculptures, maps, plans, and most recently computer programs. Section 2 of the *Copyright Act* classifies computer programs as literary works for the purposes of copyright protection. Providing copyright protection to a work allows the copyright holder the right make copies of the work, and to prohibit others from making copies.[75] [76] As with copyright in the days of the *Statute of Anne*, copyright holders may freely licence or assign their economic rights.[77]

a) The Idea/Expression Dichotomy

Copyright protects the expression of ideas, but does not grant protection to the ideas themselves.[78] This separation is referred to the idea/expression dichotomy.[79]

---

75 Barry B. Sookman, *Computer Law: Acquiring and Protecting Information Technology* (Toronto: Carswell, 1989), at p. 3-1.

76 The "essential characteristic [of copyright] is the sole right to produce or reproduce any such work or any substantial part thereof in any material form whatsoever", H.G. Fox, The Canadian Law of Copyright and Industrial Designs 2nd edition, (Toronto: Carswell, 1967), at p. 2.

77 Copyrighted works consist of two components: economic rights and moral rights. The former refers to the rights of the copyright holder to reap economic benefits for authorizing use of the work, whereas the latter refers to the author's, as opposed to the copyright holder's, right to the integrity of the work as well as the right to be associated with the work "in certain circumstances" (*Copyright Act*, s. 14.1(1)). Moral rights may not be assigned but may be waived in whole or in part (*Copyright Act*, s. 14.1(2)).

78 In *Moreau v. St.Vincent*, [1950] Ex. C.R. 198, Thorson P. stated,

Finding the line that delineates idea from expression is not an easy task, and is even more pronounced when dealing with computer programs which are, by their very nature, utilitarian works and hence intertwined with the ideas they seek to express.[80] [81] The difficulty inherent in creating a test that distills expression from idea is that, with an

---

> [A]n elementary principle of copyright law [is] that an author has no copyright in ideas but only in his expression of them. The law of copyright does not give him any monopoly in the use of the ideas with which he deals or any property in them, even if they are original. His copyright is confined to the literary work in which he has expressed them. The ideas are public property, the literary work is his own. Every one may freely adopt and use the ideas but no one may copy his literary work without his consent. (At p. 203).

79 This principle is fundamental to copyright law and has been well documented by the case law. See: *Apple Computer Inc. V. Mackintosh Computers Ltd.* (1986), 10 C.P.R. (3d) 1 (F.C.T.D.); varied (1987), 44 D.L.R. (4th) 74 (Fed. C.A.), aff'd [1990] (S.C.C.), per Reed J. (F.C.T.D.); *Delrina Corp. v. Triolet Systems Inc.* (1993), 47 C.P.R. (3d) 1 (Ont. Ct. Gen. Div.); *Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.* 797 F.2d 1222 (3rd Cir. 1986); *Computer Associates International, Inc. V. Altai, Inc.* 23 USPQ2d 1241 (2nd Cir. 1992); and *Autodesk Australia Pty Ltd. v. Dyason* (1990), 18 I.P.R. 109 (Aust. Fed. Ct.), reversed (1992), A.I.P.C. 90,855 (Aust. H.C.).

80 In *Delrina Corp. v. Triolet Systems Inc.*, supra, note 14,O'Leary J. borrowed from the decision in *Computer Associates International, Inc. V. Altai, Inc.*, where Walker J. stated,

> [d]rawing the line between idea and expression is a tricky business. Judge Learned Hand noted that "[n]obody has ever been able to fix that boundary, and nobody ever can." *Nichols*, 45 F.2d at 121. Thirty years later his convictions remained firm. "Obviously, no principle can be stated as to when an imitator has gone beyond copying the 'idea' and has borrowed its 'expression,'" ...
>
> The essentially utilitarian nature of a computer program further complicates the task of distilling its idea from its expression.

Supra, note 52, at p. 4735.

81 The doctrine of merger holds that if an expression is necessary to the function or efficiency of that idea, the component is considered necessarily incidental to the idea and is not protectable as an expression. Such a form of expression is said to be purely functional and the idea and expression merge. As the idea is inextricably linked with the expression, such expression is not protected. For example, where a program requires the user to type the word 'print' followed by the command 'full page' in order to direct the output of a program to a printer, the command sequence will not be protected as it is necessarily incidental to the idea of printing.

Related to the doctrine of merger is the doctrine of *scènes a faire*. *Scènes a faire* holds that where elements of a work are necessarily incorporated into the expression of a work, not because

overly liberal view of expression, one risks granting monopoly protection to the first authors of programs that perform certain tasks,

> [which] would thereby inhibit other creators from developing improved products. [Conversely, d]rawing the line too conservatively would allow programmer's efforts to be copied easily, thus discouraging the creation of all but modest incremental advances.[82]

In order to devise a sound test,

> the court must be faithful to the statutory language and mindful of both the ultimate goal of copyright law – the advancement of public welfare – and Congress' chosen method of achieving this goal – private reward to the individual author.[83]

The consequences of an imperfect test can illustrated by examining the decision in *Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.* and its short-lived but highly controversial legacy.[84] In terms of the idea/expression dichotomy, the *Whelan* court decided that, "the purpose or function of a utilitarian work would be the work's idea, and everything that is not necessary to that purpose or function would be part of the

---

of creativity of selection and expression but because of external factors, there will be no copyright protection granted.

82 *Supra*, note 31, at p. 53, quoting from Peter S. Menell, "Scope of Copyright Protection for Programs", 41 Stan.L.Rev. 1045, at pp. 1047-48.

83 *Supra*, note 31, at p. 53.

84 In *Whelan Associates Inc. v, Jaslow Dental Laboratory Inc.*, 797 F.2d 1222 (3rd Cir. 1986), *cert. denied* 479 U.S. 1031 (1987), a medical software developer (the defendant), after developing a dental laboratory program for the plaintiff, started up her own company and developed a similar dental lab program (using a different computer language). The Third Circuit Court of Appeals found that the defendant's program copied the structure, sequence and organization of the plaintiff's program and that this was enough to constitute an infringement of the plaintiff's copyright. This decision was revolutionary in that it formally extended software copyright protection beyond the literal copying of source code to non-literal elements, in this case the structure, sequence and organization of the program. The *Whelan* court held that since the plot of a story or play is protected by copyright so, therefore, should the sequence and organization of programs. In its decision the court reasoned that there were many possible ways in which to organize the idea of a dental lab program, and therefore the particular way which the plaintiff chose was a copyrightable expression of that idea.

expression of the idea."[85] The *Whelan* test was subsequently criticized as having "a somewhat outdated appreciation of computer science"[86] and for ignoring "practical considerations".[87] In *Computer Associates International, Inc. V. Altai, Inc.*, which was decided after the *Whelan* case, the Second Circuit of the United States Court of Appeals devised a superior three part, abstraction - filtration - comparison test, that would better distill idea from expression in computer programs.[88] According to Walker J. of the *Altai* court,

> [i]n ascertaining substantial similarity under this approach, a court would first break down the allegedly infringed program into its constituent structural parts [(abstraction)]. Then, by examining each of these parts for such things as incorporated ideas, expression that is necessarily incidental to those ideas, and elements that are taken from the public domain, a court would then be able to sift out all non-protectable [sic] material [(filtration)]. Left with a kernel, or possibly kernels, of creative expression after following this process of elimination, the court's last step would be to compare this material with the structure of an allegedly infringing program [(comparison)].[89]

The abstraction - filtration - comparison test has now gained widespread acceptance in the United States,[90] and is being used increasingly in copyright -- computer program cases abroad.[91]

---

85 *Ibid*, at p. 1236.

86 *Supra* note 52, at p. 1252.

87 *Ibid.*

88 In *CMAX v. UCR Inc.*, 4 CCH Computer Cases ¶ 46,752 (U.S. Dist. Ct., Ga., 1992) Fitzpatrick D.J. stated that the decision in *Whelan Associates Inc. v, Jaslow Dental Laboratory Inc.* is "conceptually overbroad and descriptively inadequate", preferring instead to follow the three-part test outlined in *Computer Associates International Inc. v. Altai Inc.*.

89 *Supra*, note 52, at pp. 1252 - 53. An earlier manifestation of the abstraction - filtration - comparison test was recommended by David Nimmer *et al.* in "A Structured Approach to Analyzing the Substantial Similarity of Computer Software in Copyright Infringement Cases", 20 Ariz. St. L.J. 625 (1988); 3 Nimmer & Nimmer, Nimmer on Copyright, § 13.03[F] at 13 - 78.26.

90 See *CMAX v. UCR Inc.*, 4 CCH Computer Cases ¶ 46,752 (U.S. Dist. Ct., Ga., 1992); and *Lotus Development Corporation v. Borland, Inc.*, 788 F.Supp. 78 (D. Mass. 1992).

b) Formalities, Term and Ownership

In order to avail itself of copyright protection a work may be either published or unpublished.[92] Under Canadian copyright law there is no registration[93] or marking requirement[94] for a work to be copyrightable; copyright is said to subsist upon the creation of the work.[95] The term for which copyright subsists in the work is the life of the author, or in the case of joint authorship of a work,[96] the longest surviving author, plus fifty years.[97] The author of a work is presumed to be the first owner of the copyright therein, except where a work is created under a contract of service wherein the employer is presumed to be the first owner of the work.[98]

---

91 The abstraction - filtration - comparison test has been adopted into Canadian law by the decision in *Delrina Corp. v. Triolet Systems Inc.* (1993), 47 C.P.R. (3d) 1 (Ont. Ct. Gen. Div.); and into U.K. law in *John Richardson Computers v. Flanders and Chemtec* (U.K. High Court, February 19, 1993).

92 *Copyright Act*, s. 3.

93 Although it is not mandatory that one register one's copyright in the work, it is considered prudent to register a work for several reasons. First, registration provides an evidentiary record of the work should a dispute as to its authorship ever arise, and second the copyright holder may have a broader range of remedies available should there be an infringement of the copyright. S.39 of the *Copyright Act* states that where a work is not registered, and the infringer alleges no knowledge of copyright in the work, the copyright holder is only entitled to injunctory relief unless he can prove knowledge of copyright on the part of the infringer. In cases where the work is duly registered with the Copyright Office, s.39 deems the infringer to have had knowledge of the work's copyright; consequently, the infringer may also be held liable for damages, or any other remedy that may be available. Sunny Handa and James Buchan, "Copyright as it Applies to the Protection of Computer Programs in Canada", (1994) I.I.C. [pending publication], at pp. 3 - 4. If a computer program copyright is registered, there is no requirement that the copyright holder file the detailed source code specification with the copyright office. In fact the Canadian Copyright Office will not accept attachments when registering the copyright in a work.

94 Canada is a long standing signatory to the *Berne Convention* on Copyright (Rome Revision, 1928) which prohibits any requirement that works be registered or that the "©" symbol be used in conjunction with expressions of the work in order for copyright protection to apply. *Ibid*, at p. 3.

95 *Copyright Act*, s. 5.

96 A work joint authorship is said to exist where the "contribution of one author is not distinct from the contribution of the other author or authors." *Copyright Act*, s. 2.9.

97 *Copyright Act*, s. 6 & s. 9.

98 *Copyright Act*, s. 13(3). A "contract of service", which denotes an employment relationship in the tradition sense, is to be differentiated from a "contract for services", which refers to a

c) *Originality*

As mentioned, copyright applies to protect the expression of an idea while not protecting the underlying idea itself. However, merely expressing oneself in one of the protected forms enumerated above may not in itself be sufficient to obtain copyright protection. The *Copyright Act* requires that protectable works, at a minimum, demonstrate a modicum of originality.[99] Originality in a work refers to the degree of the author's creative or inventive thought, and is comparatively low in common law copyright jurisdictions as compared with continental *droit d'auteur* jurisdictions such as Germany or France.[100] Effectively, under common law copyright systems, in order to demonstrate originality one need only show that the work originated from the author and was not a copy of an existing work.[101] Under the German copyright regime, a work must

---

independent contractor who has arranged to produce a work under a specific contract as opposed to a general employment contract. In either case the presumption created is rebuttable at law. See, *Orbitron Software Design Corp. v. M.I.C.R. Systems Ltd.* (1990), 48 B.L.R. 147 (B.C.S.C.); *Positron Inc. v. Desroches*, [1988] R.J.Q. 1636 (Que. Superior Ct.); and *Lamb v. Evans*, [1893] 1 Ch. 218 (C.A.).

99 *Copyright Act*, s. 5.

100 *Supra*, note 93, at p. 4.

101 According to the Court in *University of London Press Ltd. v. University Tutorial Press Ltd.* [1916], 2 Ch. 601 at 608,

> The word 'original' does not in this connection mean that the work must be the expression of an original or inventive thought, copyright acts are not concerned with the originality of ideas, but with the expression of thought, and in the case of 'literary work', with the expression of thought or writing. The originality which is required relates to the expression of the thought.

Similarly, in *Delrina Corp. v. Triolet Systems Inc.*, O'Leary J. stated that for a work to be considered original, "it must not have been copied by the author from another work, whether that work was protected by copyright or was in the public domain and free for the taking." (*Supra*, note 14, at p. 32).

Once originality has been demonstrated, the amount of artistic merit that must be present in the work is minimal. In *Cardwell v. Leduc* (1962) 23 Fox Pat. C. 99 (Ex.Ct.), the Court ruled that,

display a high degree of creativity (*Gestaltunghöhe*) and individuality (*Individualität*),[102] whereas under the French regime a lesser degree of originality amounting to "the evidence of an intellectual contribution of the author" and "the novel nature of the program" as compared with existing programs need be shown.[103]

d) Fixation

Additionally, in order to be worthy of copyright protection, a work must be fixed, or stored, in some manner. The fixation requirement has developed largely through copyright jurisprudence and is only statutorily based for dramatic works, musical works, and most recently for computer programs. For works where fixation is not explicitly required by the *Copyright Act*, it had been inferred as existing by the courts.[104] For computer programs, section 2 of the *Act* requires that a computer program be "expressed, fixed, embodied or stored in any manner" in order for copyright protection to apply. The U.S. *Copyright Act* is slightly more specific in that it requires that a work be expressed in a form which is sufficiently permanent and stable so that it may be

---

> Assuming for a moment that originality is conceded, I think, particularly as literary merit need not be of high order, the plaintiff's composition discloses at least a modicum of literary merit attributable to his skill and ingenuity. This added to the considerable time, care, and effort which he devoted to it, in my opinion, is more than sufficient to endow the plaintiff's [work] with the quality of "a literary work" as defined in the foregoing s.2(n).

102 Clifford Chance, "The European Software Directive" (Clifford Chance, U.K. 1991), at p. 22.

103 *Ibid*, at p. 19.

104 In *Canadian Admiral Corp. Ltd. V. Rediffusion Inc*, [1954] Ex.C.R. 382, the Court ruled that "for copyright to subsist in a 'work' it must be expressed to some extent at least in some material form, capable of identification and having a more or less permanent endurance." (At p. 394).

"perceived, reproduced, or otherwise communicated for a period of more than transitory duration, either directly or with the aid of a machine or device."[105]

Issues of fixation with respect to computer programs are many, and without further legislative or juridical guidance a good deal of uncertainty continues to exist. An example of a problematic question that had existed was whether literary and artistic works, such as text and graphic output screens, which can only be displayed during program execution will be regarded as being fixed in the memory devices which contain the computer programs and data. The recent jurisprudence with respect to the protection of non-literal elements, such as computer screens, suggests that fixation in volatile memory devices such as a video interfaces RAM is sufficient to meet the fixation requirement.[106] In the case of computer screens, the protection exists because the screens are said to exist under the umbrella of the underlying computer program's copyright.[107] A more difficult question might be whether transitory combinations of data, such as the results of a database search conducted at the direction of a user, are sufficiently fixed for the purposes of copyright since these results are often only stored in volatile memory, and cannot be said to be part of the underlying search engine as the user's search criteria is entered only upon use.

---

105 17 U.S.C. § 102.

106 *Supra*, note 14.

107 A fleeting image of broadcast on television has been held not to fulfill the requirements of fixation for the purposes of copyright in itself. If the television program is otherwise fixed, such as on some form of video tape, then the fleeting image described may be protected as part of the underlying copyright of the fixed program. *Supra*, note 104.

In terms of the reverse engineering debate, the issue of fixation is important since in order to reverse engineer a computer program one must first "copy" that program into a computer's RAM memory for a disassembler to work. As the disassembler performs its passes through the program it seeks to disassemble, it continually makes increasingly precise translations of the program which are stored in RAM.[108] Once the disassembler completes its disassembly, the results are usually stored on more permanent media such as a disk, and in print-out form. It is, however, possible to disassemble a program, or parts thereof, without storing either the intermediate copies or final result of the disassembly on any media other than in RAM. Although this may seem impractical since the results will remain within the dynamic RAM of the computer, it may provide a technical way around the difficulties of copyright infringement since without adequate fixation, copyright may not consider that any copy has indeed been made. Furthermore, if it is deemed that, even where the final product of disassembly is fixed, the intermediate copies stored in RAM do not constitute infringing copies for want of fixation, the exception to copyright found in section 27(2)(l) of the *Act*, discussed in the following section, may apply.[109]

---

108 Depending on the disassembler and the size of the program code being disassembled, these intermediate translations may also be temporarily stored on disk. Once on disk the issue of fixation becomes moot.

109 Section 27(2)(l) allows a *single* reproduction of an authorized copy of a computer program to be made where the purpose of the reproduction is the modification of the program for purposes of compatibility. In order to use that section to exempt disassembly from infringing copyright, as only a single reproduction is contemplated by the section, the intermediate copies would have to somehow be exempt from copyright. Failing to qualify as fixed would be one was of ensuring this. See *infra*, section III. A. 1. f. (1) Translation - Modification Exception for a discussion of this exception.

e) Infringement of Copyright

Once copyright subsists in a work, it will be infringed by "any person who, without the consent of the owner of the copyright, does anything that by th[e *Copyright Act*] only the owner of the copyright has the right to do.[110] It must be stressed that copyright only prevents the copying of a work, or a substantial part thereof;[111] the monopoly rights granted by the *Copyright Act* do not extend to situations where a person independently creates a similar work. Accordingly, the jurisprudence that has developed regarding infringement holds that the onus to demonstrate copying rests on the plaintiff who must demonstrate both substantial similarity between his/her work and that of the defendant, as well as a causal connection between the two works. [112] The causal connection element can be satisfied by demonstrating that the defendant had access to the original work. Once the plaintiff has discharged his/her burden by demonstrating, a rebuttable presumption is created whereby the onus in demonstrating independent

---

110 *Copyright Act*, s. 27(1).

111 By "substantial" we do not mean a strict percentage; instead, "substantial" refers to the quality of the part taken. In *Breen v. Hancock House Publishers Ltd. et al.* (1986), 6 C.I.P.R. 129 (F.C.T.D.), it was held that,

> Although the proportion of cribbing from the plaintiff's work to the total of the author's was quantitatively small, the quantitative aspect indicated to me that it was more than a 'fair deal' and that it constituted an appropriation by the author of the skill, and time, and talent of the plaintiff. As a result, the Plaintiff was entitled to an injunction. (At p. 133).

Similarly, in *SAS Institute Inc. v. S & H Computer Systems* 605 F.Supp. 1816 (U.S.D.C. 1978), the court found that 44 examples of copying had occurred out of a total of approximately 186,000 lines of computer source code. The court held that these 44 examples of copying constituted a substantial taking and that simply because there were only 44 instances of copying did not necessarily mean that the copying was trivial. (At p. 822).

112 *Gondos v. Hardy* (1982), 64 C.P.R. (2d) 145 (Ontario. H.C.); *Francis Day & Hunter Ltd. V. Bron*, [1963] Ch. 587 (C.A.).

creation shifts to the defendant. The defendant may also attempt to rely on statutory exceptions to infringement contained in section 27(2) of the *Copyright Act*.

f) Exceptions to Infringement

(1) Translation - Modification Exception

If the intermediate copies or final results of the disassembly process are indeed considered to be fixed, thus nullifying any argument that copyright doesn't apply because of lack of fixation, the copying of a computer program for use by a computer *may* be exempted by section 27(2)(l) of the *Copyright Act*. Whether such use includes disassembly is the crucial point. Section 27(2)(l) states that infringement does not occur as a result of,

> the making by a person who owns a copy of a computer program, which copy is authorized by the owner of the copyright, of a single reproduction of the copy by adapting, modifying or converting the computer program or translating it into another computer language if the person proves that
>
> (i) the reproduction is essential for the compatibility of the computer program with a particular computer,
> (ii) the reproduction is solely for the person's own use, and
> (iii) the reproduction is destroyed forthwith when the person ceases to be the owner of the copy of the computer program [113]
> ...

In drafting this section "[t]he House of Commons Sub-Committee on the Revision of Copyright recognized that it is common in the industry for computer programs to be

---

[113] Computer programs, however, are generally licensed and are only rarely sold by the original copyright holder. Whether a statutory provision, which requires ownership of the software, may actually be enforced in licensing situations remains questionable. See *infra*, note 119, and accompanying text for a discussion of licensing, as contrasted with the sale, of a software product.

adapted or modified to meet the particular needs of ends users."[114] Accordingly, this section would avoid such modification from being an infringement. Disassembly was not contemplated as being within the scope of the section by the Sub-Committee.

This section may, in fact, also be used to support a claim that in order to function, computer programs must be copied into parts of a computer (usually into RAM), and translated by the central processing unit into microcode in order to run as intended.[115] Although it can be argued that section 27(2)(l) protects against a computer program's use being declared as infringing, it is more tenuous to argue that this section also applies where a program is copied into RAM and subsequently onto more permanent media for the purposes of dissection by a disassembler, a necessary step in the disassembly of a computer program.[116] However, where simple use of a program is

---

114 *Supra*, note 75, at p. 3-203.

115 A similar exception was placed into the U.S. *Copyright Act* in 1988. The purpose of 17 U.S.C. §117 was to allow authorized users the right to use a computer program without technically infringing the copyright in the program. The section reads:

> Notwithstanding the provisions of section 106 [17 USCS Sect. 106], it is not an infringement for the owner of a copy of a computer program to make or authorize the making of another copy or adaptation of that computer program provided:
>
> (1) that such a new copy or adaptation is created as an essential step in the utilization of the computer program in conjunction with a machine and that it is used in no other manner ...

The U.S. exception, however, has been interpreted as not applying to the reverse engineering of computer programs: *Sega Enterprises v. Accolade Inc.*, 24 U.S.P.Q.2d 1561 (9th Cir. 1992), at p. 1568. See *infra*, note 260 and accompanying text.

116 According to one commentator,

> The right [under section 27(2)(l)] to convert a computer program or translate it into another computer language will probably give a person who owns an authorized copy of a computer program the right to convert the program from one higher-level language to another. It

concerned, even if section 27(2)(l) does not apply, it is unlikely that the copying of the program for the purposes of use will be declared as infringing for the simple reason that the conversion of object code into electrical signals may not be a "reproduction in material form".[117]

This technical difficulty with the simple use of an authorized copy of a computer program, in its intended manner, potentially constituting copyright infringement provides a good illustration of the inappropriateness of copyright as the principal form of intellectual property protection for computer programs.[118]

(2) Making Backup Copies

In addition to the translation/adaptation/modification, contained in section 27(2)(l) of the *Act*, there are two other exceptions to infringement that also apply to computer programs. The more specific of these exceptions is contained in section 27(2)(m) of the *Copyright Act* and authorizes the owner of an authorized copy of a computer program to make a single copy for backup purposes, however, this copy must be destroyed as soon as the person ceases to be the owner of the copy. Practically speaking, this provision, as well as that contained in section 27(2)(l), have proven to be

---

> might also give such a person the right to convert a program from a higher-level language to machine language, and vice-versa, but the meaning of the term 'translation' in the Act is still uncertain ... and so the scope of [27(2)(l)] ... is still not known. *Supra*, note 75, at p. 3-204 [emphasis added]. See *infra*, note 143 and accompanying text.

117 *Supra*, note 18, per Mahoney J.A. (Fed.C.A.).

118 *Supra*, note 75, at p. 3-3, note 17 and accompanying text.

of limited use as software companies seldom transfer the ownership of their software, preferring instead to license it to users. Licensing effectively allows the software companies to supersede certain provisions of the *Act*, such as this backup exception to infringement, with the terms as contained in their licensing agreement.[119] Most software companies do, nonetheless, permit the making of a backup copy in their licensing agreement as it saves them the headache of replacing programs where the carrying media becomes defective.[120]

Licensing terms can also be used to prevent the reverse engineering of the computer program by the user. If the user is a licensee, rather than the *owner* of a copy of the computer program, then *prima facie* the licence terms will prevail notwithstanding whether reverse engineering is permitted under the *Copyright Act*. If a licence is silent, or ambiguous, as to particulars regarding interpretation then the court will gap-fill using

---

119 As a result of the mass production of off-the-shelf software, it has become quite impossible for software publishers who deal in such software to individually negotiate licence terms with each prospective purchaser. In order to combat this problem, "shrink-wrap" licenses were developed. Shrink-wrap licences consist of a list of licensing terms which a licensor places visibly on a product (usually under the cellophane wrapper) for a prospective purchaser to read. Typically, the licence also has a clause which states that if you do not agree to abide by the terms of the licence then you should not purchase the product, and that by opening the packaging you are agreeing to abide by the terms. It is not clear whether shrink wrap licences are enforceable in Canada even through they are an extremely common industry practice. In an often cited quote from *Betts v. Wilmott* (1871), 6 Ch.App. 239 (U.K.C.A.), the Court held that "when a man has purchased an article he expects to have the control of it and there must be some clear and explicit agreement to the contrary to justify the vendor in saying that he has not given the purchaser his licence to sell the article, or to use it wherever he pleases, as against himself". In *North American Systemshops Ltd. v. King* (1989), 68 Alta. L.R. (2d) 145 (Q.B.D.), the Court found that a shrink-wrap licence agreement contained within the packaging of a software product, and not visible to the purchaser at the time of purchase, was not enforceable. Whether or not the placing of shrink-wrap licences on the outside of packaging fulfills the requirement of a "clear and explicit agreement" remains to be seen.

120 Unlike the Canadian and U.S. *Copyright Acts*, similar backup copy provisions contained in the U.K.'s *Copyright, Designs and Patents Act 1988* cannot be contracted out of. See *infra*, note 297, and accompanying text.

implied terms which are reflective of industry practice.[121] Licence terms that run afoul of the *Copyright Act* have not been tested in Canadian courts. The experience in the U.S., however, would suggest that such terms may not be enforceable.[122]

(3) Fair Dealing Under the *Copyright Act*

Another copyright exception, which is general as opposed to computer program specific, however, is the fair dealing exception found in section 27(2) of the *Copyright Act*. According to that section, "any fair dealing with any work for the purposes of private study [or] research ..." will not constitute an infringement of copyright in the work. The breadth of the fair dealing exception is not further detailed in the *Act*, and has only been interpreted by a smattering of jurisprudence. Similar exceptions also exist in U.K. copyright legislation, where the term "fair dealing" is also used, and, more importantly for the purposes of reverse engineering computer programs, in the U.S. copyright legislation where the term "fair use" is used. As a result of the paucity of Canadian jurisprudence on the subject of fair dealing, it is impossible to say how similar our exception will be to those of other jurisdictions.[123] As with any other

---

121 The theory behind implied licences is that no two parties to a transaction would enumerate all of the terms possible that relate to their relationship – to do so would be extremely costly. Instead, the parties only agree on those terms which are unconventional to industry practice. It is assumed that the parties intentionally remained silent about all of the other terms as they were reflective of industry standards -- this would allow the parties to cut down on contracting (transaction) costs. The difficulty with implied licences in the computer software, and any other new industry, is that industry practices often are not fully established when a court is forced to decide upon a relationship where the terms have not been expressly spelled out.

122 See *infra*, at p. 52, for a detailed discussion of trade secrets and licensing.

123 It is widely thought that "fair dealing" and "fair use" are different. With respect to the home taping issue decided by the U.S. Supreme Court in *Universal City Studios v. Sony Corporation of America*, 220 U.S.P.Q. 665 (1984), according to one commentator, "there are sufficient differences between the American "fair use" defence and Canada's "fair dealing" to conclude that, if an action were brought in Canada, home taping would be found to constitute a copyright

statute that has been interpreted to a limited degree, use of similar legislation in other jurisdictions is useful yet not binding.[124] In the context cases "related to copyrightability of computer programs American authorities have been cited and, notwithstanding the differences between the wording of the Acts, have been given qualified approval."[125] No court has as of yet performed a comparative analysis of the Canadian "fair dealing" and the American "fair use" exceptions. This, however, is not the case where the U.K. fair dealing exception is concerned. As the Canadian *Act* was borne out of the Imperial Statutes of the same name, much of the early Canadian copyright jurisprudence was borrowed from the U.K.. The few Canadian courts that have dealt with fair dealing cases have extensively borrowed from their U.K. counterparts.

The earliest Canadian case to deal extensively with the fair dealing exception was *Zamacois v. Douville*.[126] The Exchequer Court in that case laid out the basic principles that govern fair dealing as: a verdict of fair dealing must depend on the specific facts of each case; the copying of an entire work cannot qualify as a fair dealing; short of copying the entire work, the quantity of the work copied is not solely determinative of fair dealing; and "in considering whether a dealing with a particular work

---

infringement. Monique Hebert, *Copyright Act Reform*, Research Branch, Library of Parliament, Government of Canada, 1982 reviewed 1987, at p. 11.

124 See *Supra*, note 5. In accepting the value of U.S. copyright law in the field of computer software protection, O'Leary J., in *Delrina Corp. v. Triolet Systems Inc.*, *supra*, note 14, further added that, "[b]ecause computer and computer software manufacturers are concentrated in the United States, it is not surprising that U.S. courts have had to frequently deal with and have developed rules for determining disputes like the one now before me." (At pp. 32 - 33).

125 *Supra*, note 75, at p. 3-6.

126 *Zamacois v. Douville* (1943), 3 Fox Pat. C. 44 (Ex. Ct.).

[i]s fair, it would have to be considered whether any competition [i]s likely to exist between the two works."[127] The other two Canadian cases to deal with the a defence of fair dealing each faced the question of whether an abridgment, or summary, of a work could in itself avoid infringing the work through a claim of fair dealing.[128] In both cases the court held that merely summarizing a work without the addition of some further comment is not fair dealing.

In *Hubbard v. Vosper*[129] the U.K. Court of Appeal stated,

> it is impossible to define what is 'fair dealing'. It must be a question of degree. You must consider first the number and extent of the ... extracts. Are they altogether too many and too long to be fair? Then you must consider the use made of them. If they are used as a basis for comment, criticism or review, that may be a fair dealing. If they are used to convey the same information as the author, for a rival purpose, that may be unfair. Next, you must consider the proportions. To take long extracts and attach short comments may be unfair. But, short extracts and long comments may be fair. Other considerations may come to mind also. But, after all is said and done, it must be matter of impression.

In *Beloff v. Pressdram*[130] the U.K. Chancery Court added another element to the fair dealing defence: fair dealing in light of unpublished and confidential information. In that case, the defendant, a newspaper company, published an unpublished internal office-memorandum written by the plaintiff, without obtaining the plaintiff's authorization to do so. The defendant claimed that its purpose in publishing the plaintiff's

---

127 *Ibid*, Annotations at pp. 72 - 4.

128 *Breen v. Hancock House Publishers* (1985), 6 C.I.P.R. 129 (F.C.T.D.), at p. 133; and *R. V. James Lorimer and Co. Ltd.*, [1984] 1 F.C. 1065 (F.C.A.), at pp. 1077 - 78.

129 *Hubbard v. Vosper*, [1972] 1 All E.R. 1023 (C.A.), at p. 1024.

130 *Beloff v. Pressdram*, [1973] 1 All E.R. 241 (Ch. D.).

work was to criticize it and that it was covered by the fair dealing exception under the *Copyright Act*. The court held that the fact that the memorandum had not been published was not in itself enough to find in favour of the plaintiff; however, it was an aggravating factor to be taken into account in assessing the defendant's conduct. With respect to the confidentiality of the information used, the court held that,

> [t]he vice of the leak of the publication in this case was, to my mind, clearly unjustifiable for the authorised purposes of criticism, review and news, and clearly in my view constituted dealing which was not fair within the statute. ... This ground is ample to defeat the defence of fair dealing ...[131]

Each of the aforementioned dealt with fair dealing in the context of traditional literary works. No case in Canada or the U.K. has applied the defence of fair dealing in the context of a computer program. The American experience with their fair use exception has been quite different. The tests for fair use are highly evolved, dealing with all sorts of subject matter, and two of the most recent cases have directly applied fair use to computer program works.[132] The American fair use exception will be discussed in further detail in Chapter IV, where the rulings in these cases will also be analyzed.

(4) Public Interest Exception

A final exception to copyright infringement in reverse engineering cases, if the exception indeed exists, is the public interest defence. This defence is judicially created, and does not expressly appear in Canadian copyright legislation.[133] The public

---

131 *Ibid*, at p. 264.

132 *Sega Enterprises Ltd. v. Accolade Ltd.*, 24 U.S.P.Q.2d 1561 (9th Cir. 1992); and *Atari Games Corp. v. Nintendo of America Inc.*, 975 F.2d 832 (Fed. Cir. 1992).

133 Section 171(3) of the U.K.'s *Copyright, Designs and Patents Act 1988*, c. 48, expressly recognizes the existence of a public interest defence and states that "[n]othing in this Part affects

interest defence has not been successfully raised in Canada with respect to copyright, and has only gained judicial recognition in a small passage in one decision: *The Queen v. James Lorimer & Co. Ltd.*.[134] In that case, Mahoney J. stated that,

> I have no doubt that a defence of public interest as enunciated in the English cases is available in proper circumstances against an assertion of Crown copyright. ... [however t]his is not a 'public interest' case in the same sense as the English decisions nor, really, in the sense the defence was advanced here.[135]

The English cases Mahoney J. was referring to were: *Hubbard v. Vosper*[136] and *Beloff v. Pressdram*.[137] In both of those cases, the courts recognized that a common law defence of public interest was available notwithstanding its lack of legislative mention. In this regard, Ungoed-Thomas J. stated that the defences of public interest and fair dealing "are separate defences and ... are governed by separate considerations. Fair dealing is a statutory defence limited to copyright infringement only. But public interest is a defence outside and independent of statutes, is not limited to copyright cases and is based on a general principle of common law."[138]

The discussion of a public interest defence in the context of a copyright infringement claim was further examined in *Lion Laboratories Ltd. V. Evans*.[139] In that case, the U.K. Court of Appeal was faced with whether the theft of confidential literature

---

any rule of law preventing or restricting the enforcement of copyright, on grounds of public interest or otherwise."

134 *R. v. James Lorimer & Co. Ltd.*, [1984] 1 F.C. 1065 (F.C.A.).

135 *Ibid*, 1078.

136 *Supra*, note 129.

137 *Supra*, note 130.

138 *Ibid*, at p. 259.

139 *Lion Laboratories Ltd. V. Evans*, [1985] Q.B. 526 (C.A.).

which was subsequently published, without the copyright holder's permission, could be held to be in the public interest. The literature in that case was a confidential internal memorandum detailing information that a breathalyzer device, manufactured by the plaintiff, was capable of giving false readings which may have been responsible for the conviction of innocent persons. The defendants did not deny that they took the confidential information without the plaintiff's permission or that their publication did not *prima facie* infringe the plaintiff's copyright; the only defence presented was one of public interest In deciding how to weigh the public interest against the copyright infringement and breach of confidence perpetrated by the defendants, Stephenson L.J. stated,

> "[t]o be allowed to publish confidential information, the defendants must do more than raise a plea of public interest; they must show 'a legitimate ground for supposing it is in the public interest for it to be disclosed' ... we 'should not restrain it by interlocutory injunction, but should leave the complainant to his remedy in damages.[140]

Griffiths L.J. agreed, and in assessing the applicability of the public interest defence to copyright infringement stated,

> I am quite satisfied that the defence of public interest is now well established in actions for breach of confidence and, although there is less authority on the point, that it also extends to breach of copyright: see by way of example *Fraser v. Evans* [1969] 1 Q.B. 349; *Hubbard v. Vosper* [1972] 2 Q.B. 84; *Woodward v. Hutchins* [1977] 1 W.L.R. 760 and *British Steel Corporation v. Granada Television Ltd.* [1981] A.C. 1096. ... When there is an admitted breach of confidence and breach of copyright, there will usually be a powerful case for maintaining the status quo by the grant of an interlocutory injunction [;] ... the court must appraise it critically; but if convinced that a strong case has been made out, the press should be free to publish, leaving a the plaintiff to his remedy in damages.[141]

---

140 *Ibid*, at p. 538.
141 *Ibid*, at p. 550.

Although the jurisprudence dealing with public interest is solidly in place in the U.K., in Canada the sole mention of a public interest copyright defence in *The Queen v. James Lorimer & Co. Ltd.* expressly mentioned the existence of such a defence in the context of *Crown* copyright. It would, however, be non-sensical if the defence does not also extend to defend against claims by private copyright holders. As it stands, a public interest defence seemingly exists in Canada, independent of any statutory exception to copyright.[142]

Both the fair dealing exception and/or the public interest defence, it will be shown, are essential to an argument in support of reverse engineering which is most probably an infringement of copyright under the Canadian legislation as it exists today. With respect to the former, the related fair use exception, which appears in the U.S. *Copyright Act*, has been successfully used in reverse engineering cases, and notwithstanding the differences between the respective exceptions, fair dealing remains the most likely candidate to allow the reverse engineering of computer programs under Canadian copyright law (unless otherwise amended). If the fair dealing defence fails to support an argument allowing reverse engineering then a more tenuous, although nonetheless plausible claim, may be made under the principle of public interest.

---

142 Although section 63 of the *Copyright Act* effectively abolishes common law copyright, it makes no mention of common law defences to copyright infringement. The existence of an implied public interest defence may be rooted in the modern day general under-pinnings of copyright law protections which seek to balance the public's right to knowledge with the individual's right to be remunerated for his/her work. *Supra*, note 75, at p. 3-1, note 2 and accompanying text.

Copyright, however, may not be the sole hurdle to the reverse engineering of computer programs. Additional protections such as those raised under trade secrets law, or by licensing provisions which seek to expand the scope of copyright protection, may also throw a road-block in front of any attempt to reverse engineer a computer program. These other protections will be discussed below.

B. *Is Reverse Engineering An Infringement Of Copyright Law?*

With the fundamental principles in place, one is, at this point, inclined to ask whether reverse engineering is indeed an infringement of copyright law, and if so, how? As mentioned, when reverse engineering a computer program the disassembler must first load a copy of the program into a computer's memory.[143] This is the first potentially infringing copy. As the disassembler makes passes over the program it is seeking to dissect, it will continue to produce potentially infringing copies of the program as translations of the work, contrary to section 3(1)(a) of the *Copyright Act*. These copies are referred to as intermediate copies.[144] Once the disassembler has completed its task, it will produce an assembler language version of the computer program. This assembler source code constitutes yet another potential infringement of the computer program's copyright. Generally, the assembler will not reconstitute a program in the exact fashion in which it was written. The resulting source codes will constitute a

---

143 Although the operating system also creates a copy of a computer program in order to execute the program, this action will not be regarded as an infringement under the *Act*. See *supra*, note 116, and accompanying text; and *supra*, note 117, and accompanying text.

144 *Atari Games Corp. v. Nintendo of America Inc.*, 975 F.2d 832 (Fed. Cir. 1992), at p. 842.

translation rather than a reproduction,[145] both of which constitute a potential infringement of the original work.[146] Finally, making a hard, or printed, copy of the work for further examination of the computer program's operating principles will also potentially be an infringement.[147] The difficulty with reverse engineering a computer program lies in the fact that the program must be put into memory for decompilation. The act of reverse engineering is not in itself a violation of copyright, only the means by which reverse engineering is achieved violates copyright.[148] It is the thesis of this paper, however, that because of the need to develop standards and achieve program compatibility, reverse engineering is contemplated within the scope of the fair dealing provision in the *Copyright Act*.[149] Furthermore, it will also be argued that the production of intermediate

---

145 The distinction between a reproduction and a translation for the purposes of copyright law is irrelevant in the case of computer programs. See *infra*, note 18.

146 In the United States, the copies of the computer program made by the disassembler in final form would be termed "derivative works" under the U.S. *Act* (17 U.S.C. §103). A derivative work is defined as

> a work based upon one or more preexisting works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which a work may be recast, transformed, or adapted. A work consisting of editorial revisions, annotations, elaborations, or other modifications which, as a whole, represent an original work of authorship, is a "derivative work". (17 U.S.C. §101).

147 *Supra*, note 35, at p. 2011-12.

148 For other products, "to reverse engineer [them], it is a simple matter to buy as many examples as necessary to take apart, inspect, and test without copying anything." Clifford G. Miller, "The Proposal for an EC Council Directive on the Legal Protection of Computer Programs" 12 E.I.P.R. 347 (1990), at p. 349, as cited in *Supra*, note 35, at note 50. The idiosyncratic application of copyright law to computer program products makes the inspection of these products a potential infringement. Once again, it is apparent that copyright may not provide the optimal manner in which to protect these products.

149 The Canadian government did, in 1984-85, consider implementing, what would effectively amount to, a statutory exception allowing the reverse engineering of computer programs, in the *Copyright Act*. This proposal was rejected at the time. See *infra*, note 350, and accompanying text.

copies should be exempted from copyright through a statutory exception since the prohibition of reverse engineering falls beyond the scope of copyright protection. The rationale for these arguments will be more fully discussed in Chapter V.

C. *Reverse Engineering Under Other Legal Regimes*

1. Trade Secrets

Intertwined with the law of copyright insofar as it applies to reverse engineering is the law governing trade secrets. Trade secret law is judge made law that protects commercial confidences from being revealed.[150] Trade secret law is a form of intellectual property protection that can co-exist with other intellectual property protections, and in the case of literary, dramatic, artistic or musical matter may even protect the underlying ideas which are not in themselves copyrightable.[151] Trade secret protection covers a wider scope of informational elements than other forms of intellectual property protection. However, trade secret protection is also more limited in scope.

Canadian and British Courts have applied three general requirements for succeeding in a trade secret suit: (1) the information must have the necessary quality of confidence about it, (2) the information must have been imparted in circumstances importing an obligation of confidence (a "special relationship" must exist between the

---

150 According to the *American Restatement of the Law of Torts* (1939): "A trade secret may consist of any formula, pattern, device or compilation of information which is used in one's business, and which gives him an opportunity and advantage over competitors who do not know how to use it." This definition was accepted in Canada by Chevrier J. in *R.L. Crain Ltd. v. Ashton* [1949] 2 D.L.R. 471 (Ont. H.C.), affirmed [1950] O.R. 62 (Ont. C.A.).

151 *Q-Co. Industries Inc. V. Hoffman*, 625 F.Supp. 608 (S.D.N.Y. 1985).

parties); and (3) there must be an unauthorized use of that information to the detriment of the party communicating it.[152] The burden of proving that the information is indeed secret falls on the plaintiff.[153] It must be stressed that once the information loses its quality of secrecy it may no longer avail itself of trade secret protection. Authorized disclosure by the party who came up with the information vitiates this protection as does the information falling into the public domain.[154] Once the information is duplicated "either by legitimate independent research or in any other honest way"[155] it will lose its trade secret protection.[156]

a) The Legal Basis for Dealing With Trade Secrets

At law trade secret protection is often thought of as being based on the laws of property, contracts and/or trusts. In Canada, Sopinka J. stated that the legal basis for trade secret actions is "*sui generis* relying of each of these areas to enforce the policy

---

152 See, for example, *LAC Minerals Ltd. v. Intern. Corona Res.* (1989) 26 CPR (3d) 97 (SCC); *Software Solutions Associates v. Depow* (1989) 25 C.P.R. (3d) 129; *Coco v. A.N. Clark Ltd.* [1969] RPC 41 (Ch.D); and *Ridgewood Resources Ltd. v. Henuset* (1982) 18 Alta. L.R. (2d) 68 (Alta. C.A.), leave to appeal to SCC refused (1982) 43 N.R. 90 (S.C.C.).

153 The possessor of the secret does not have to go to unreasonable lengths to maintain secrecy (*Creditel of Canada Ltd. v, Faultless* (1977) 36 C.P.R. (2d) 88 (Ont. H.C.)). The possessor of the secret does not have to guard against unanticipated, undetectable, or unpreventable methods of discovery.(*International Corona Resources Ltd. v. LAC Minerals Ltd.* (1986) 53 O.R. (2d) 737 (Ont. H.C.)). *Supra*, note 75, at p. 4-29.

154 "A trade secret owner has no absolute power to exclude others from any particular activity. However, a trade secret gives the owner the right to prohibit acquisition of the protected secret by 'improper means' .... Moreover, the trade secret owner has no right to prohibit proper means of discovery, such as independent development, reverse engineering, or derivation from publicly available sources." Michael D. Stein, "The Importance of a Trade Secret as a Supplement to Copyright Protection of Computer Software", (Fall 1993) 12 I.P.L. Newsletter 28, at p. 29.

155 *Supra*, note 75, at p. 6-2.

156 *Breeze Corpns. v. Hamilton Clamp & Stampings Ltd.*, [1962] O.R. 29 (Ont. H.C.); *R.L. Crain Ltd. v. Ashton and Ashton Press Manufacturing Co.*, [1949] 2 D.L.R. 481 (Ont. H.C.); aff'd [1950] O.R. 62 (Ont. C.A.).

of the law that confidences be respected."[157] This law of confidentiality arises from an obligation of good faith in commercial settings or from a fiduciary relationship. Where no fiduciary relationship exists the courts will examine the relationship to see if it has a fiduciary quality about it that would require an element of confidence. For example, manufacturers and designers, licensors and licensees, joint venturers intending to do business with one another, and employees may all fall within the reach of trade secret laws. The duty of confidence arises when there is either an express or implied agreement between the parties that the information will not be disclosed. Providing explicit notice that the information is confidential is, of course, best. However, based on the relationship of the parties, the court may hold that the notice was implied.

In terms of conflicting with Canadian copyright law, section 63 of the *Copyright Act* which states that it is to be the sole source of copyright also contemplates the existence of trade secret laws that may work independently of the copyright legislation.[158] It has been suggested that with computer program secrets, maintaining

---

157 *LAC Minerals Ltd. v. International Corona Res.* (1989) 26 CPR (3d) 97 (SCC).

Aside from the common law basis for dealing with trade secret, several U.S. states have enacted trade secret statutes. Many of these states followed *The Uniform Trade Secrets Act* (U.T.S.A.), a codification of many of the common law principles, as a model. Michael D. Stein, "The Importance of a Trade Secret as a Supplement to Copyright Protection of Computer Software", (Fall 1993) 12 I.P.L. Newsletter 28, at p. 29.

158 Section 63 of the *Copyright Act* states:

> No person is entitled to copyright or any similar right in any literary, dramatic, musical, or artistic work otherwise than under and in accordance with this Act, or of any other statutory enactment for the time being in force, but nothing in this section shall be construed as abrogating any right or jurisdiction to restrain a breach of trust or confidence. [R.S., c. C-30, s. 45]

secrecy may be achieved by: restricting access to source code on a need to know basis, distributing software only in an object code format, and requiring licensees to refrain from engaging in reverse engineering the software product.[159] With respect to the reverse engineering debate, the use of trade secret laws are generally given effect through the latter mentioned licensing agreements, or contracts, between the parties.[160]

Although section 63 most certainly allows trade secret laws to supplement copyright principles, it is not clear whether trade secret laws would be paramount in a situation where the two laws directly conflict.[161] It has been suggested that, in the case of reverse engineering computer programs, the true aim of trade secret restrictions

> is not preservation of confidentiality or security against disclosure to third parties; it is foreclosure of competition. Rather than speaking to breach of trust through conversion of or failure to safeguard information entrusted in confidence, the contract term creates a competitive restraint by barring the conduct ...[162]

Clearly, an agreement whose purpose is to restrain competition, rather than protect against a breach of trust of confidence, is not contemplated by section 63 and will constitute both a misuse of copyright and anti-competitive behavior (see below for a discussion of these principles). Practically speaking, the intent of most contracts that restrain reverse engineering support all three objectives. In such a case it is not clear

---

159 George Fisk and Jane Clark, "Hardware and Software Protection in Canada", (1990) 10 Computer Law Journal 483, at pp. 497-98.

160 "A trade secret owner is only required to make reasonable efforts to protect the secret. There are no universally applicable procedures for protecting trade secrets". *Supra*, note 154, at p. 29.

161 See *infra*, note 193, and accompanying text.

162 *Supra*, note 45, at p. 623.

whether a court would look to the primary purpose of the agreement or whether an incidental objective to restrain a breach of confidence or trust will suffice to support the terms of the agreement as superseding copyright law.

Where computer programs are mass produced, the situation changes as there are no negotiated agreements. If an object code copy of computer software is released to the public without a licensing agreement, then there is no fiduciary or "special relationship" between the parties, and members of the public are free to reverse engineer the product to determine the source code. Of course, in the case of computer programs any reverse engineering would be subject to copyright laws. However, in the absence of a copyright prohibition on reverse engineering any member of the public, not in a "special relationship" (imparting an obligation of confidence) with the computer program owner, is legally free to attempt to reverse engineer a lawful copy of the computer program.[163] Generally, however, mass-marketed computer programs use shrink wrap license agreements[164] to enhance their copyright.

---

163 In *Geac Canada Ltd. v. Prologic Computer Corp. et al.*, [1987] Vancouver Registry C872594 (B.C.S.C.), the plaintiff alleged that one of the defendants, who was in a software licensing relationship with the plaintiff, had violated "its common-law duty of confidentiality" to the plaintiff by allowing another of the defendants to reverse engineer the licensed software and use the results obtained therefrom in the creation of compatible and even competing products. The Court refused the plaintiff's claim for an interlocutory injunction enjoining the latter defendants from continuing to market their software products. The Court based its ruling solely on the fact that the plaintiff was guilty of laches and acquiescence. The Court did not express an opinion on the issue of reverse engineering as it relates to trade secrets *per se*, although it did state that, broadly speaking, the claims raised "serious issues to be tried." No claim concerning a potential infringement of copyright was made by the plaintiffs, and, accordingly, the Court in no way addressed any issue related to copyright.

164 For a discussion of shrink-wrap licensing, see *Supra*, note 119.

2. Enhancing Copyright Protection Through Licensing/Contract Law

Several difficulties immediately arise with the use of shrink-wrap licences to enhance copyright protection. First, as mentioned, with mass marketed software it is unknown whether shrink wrap licensing is valid. Certainly, for the purposes of a trade secrets argument, it seems unlikely that a shrink wrap agreement will sufficiently constitute the "special relationship" required between the software developer and the purchaser for trade secret protection. The "special relationship" is more likely to exist where the licensing has been negotiated between the vendor and purchaser. Furthermore, notwithstanding the difficulties inherent in shrink wrap licences, if reverse engineering is deemed to be in the public interest under copyright legislation, trade secret protection, a creature of the common law, might not be extended to cover the reverse engineering of computer programs, especially if the purpose of the licence agreement is found to be a restraint of competition. The following three sections set out defences that may be used where a copyright holder attempts to prevent a user from reverse engineering a computer program through licensing or contract provisions, assuming that such reverse engineering is permitted under the *Copyright Act*, which is the position taken in this article.

a) Copyright Misuse Doctrine

The general question of whether contract/licensing law can indeed supersede copyright law provisions in a situation of conflict is yet another question that has yet to be comprehensively addressed by any court. Specifically, if copyright law supports a limited reverse engineering exception, it is arguable whether protection obtained through licensing can be used to override the copyright exception. Although this

problem has not faced Canadian courts at the time of this writing, American Courts have, through a number of decisions, developed a "copyright misuse doctrine" that may potentially be used to preempt the "enforcement of software license terms that prohibit reverse engineering".[165] Attempting to widen copyright law beyond its limited scope through contract would, under the misuse doctrine, render any attempt to enforce one's copyright invalid until the offending terms are purged.[166]

The copyright misuse doctrine was recently applied by the U.S. Federal Court of Appeals in *Lasercomb America Inc. v. Reynolds*[167] where the Court held that the misuse doctrine extends to render "a copyright unenforceable against any person regardless of whether they entered into a contract containing the offending term".[168] Under the ruling in that case, merely *attempting* to widen the scope of copyright protection beyond its accepted limits is a bar to its use. Although the U.S. Supreme Court has not applied the copyright misuse doctrine, it did acknowledge its existence in *United States v. Loew's, Inc.*.[169] The misuse doctrine is grounded in equity and as a result requires that claimants show clean hands in order to make use the defence.[170] The fact that the defence is equity based makes it potentially available to Canadian litigants in copyright matters.[171]

---

165 *Supra*, note 45, at p.551.
166 *Ibid.*
167 *Lasercomb America Inc. v. Reynolds*, 911 F.2d 970 (4th Cir. 1990).
168 *Supra*, note 45, at footnote 24.
169 *United States v. Loew's, Inc.*, 371 U.S. 38 (1962), at p. 83. See also, *Supra*, note 144, at p. 846.
170 *Supra*, note 144, at p. 846.
171 Principles of both common law and equity apply to actions concerning Federal Laws in Canada: *Aldrich v. One Stop Video Ltd.*, 13 C.I.P.R. 202, 17 C.P.R. (3d) 27 (B.C.S.C.).

In fact, the existence of a copyright misuse defence has implicitly been acknowledged by the Supreme Court of Canada in *Massie & Renwick Ltd. v. Underwriters' Survey Bureau Ltd.*.[172] In that case, the defendants, accused of copyright infringement, alleged that the plaintiff's acts, in withholding their plans and insurance rating schedules from the defendant, constituted "a combine and conspiracy" under both the *Combines Investigation Act*[173] and the *Criminal Code*.[174] The trial court ruled that the defendant's failed to demonstrate that the plaintiffs were guilty of acting in an anti-competitive manner and as a result it did not have to rule on the misuse defence. In *obiter*, the trial judge held that,

> [e]ven if the wrongs imputed against the plaintiffs were established in fact, I do not think that would deprive them of their right to protect their copyright; their copyrights would not perish because they had offended against another statute.[175]

On appeal, the Supreme Court of Canada upheld the trial court's ruling, however, with respect to the statement as to misuse, the Court added,

> if the plaintiffs in an action for the infringement of copyright are obliged, for the purpose of establishing the existence of, and their title to, the copyright to rely upon an agreement and that agreement constitutes a criminal conspiracy, and their title rests upon such agreement and upon acts which are criminal acts by reason of their connection with such an agreement, then it would be difficult, on

---

172 *Massie & Renwick Ltd. v. Underwriters' Survey Bureau Ltd.*, [1940] S.C.R. 218; var'g [1938] Ex. C.R. 103. A defence based on misuse has also been raised in several Canadian patent cases. See, for example, *Philco Products Ltd. v. Thermionics Ltd.*, [1939] Ex.C.R. 147; aff'd, [1940] S.C.R. 501, *Philco Products Ltd. v. Thermionics Ltd.*, [1941] Ex. C.R. 209, var'd [1943] S.C.R. 396, *RBM Equipment Ltd. v. Philips Electronic Industries Ltd.*, [1973] 1 F.C. 103, *RBM Equipment Ltd. v. Philips Electronic Industries Ltd.* (1973), 10 C.P.R. (2d) 23 (F.C.), *Amoco Canada Petroleum Co. Ltd. v. Texaco Exploration Canada Ltd.*, [1976] 1 F.C. 258, and *Eli Lilly & Co. v. Marzone Chemicals Ltd.* (1976), 29 C.P.R. (2d) 253 (F.C.T.D.); aff'd [1977] 2 F.C. 104.

173 *Combines Investigation* Act, R.S.C., 1927, c. 36.

174 *Criminal Code*, R.S.C., 1927, c. 36, s. 498.

175 *Massie & Renwick Ltd. v. Underwriters' Survey Bureau Ltd.*, [1938] Ex. C.R. 103 as cited in Richard B. Austin, "Misuse of Copyright", (1991) 8 Canadian Computer Law Reporter 53, at p. 57.

> general principles to understand how such an action could succeed
> ...[176]

The only other Canadian case to specifically mention the misuse of copyright defence was the decision in *Bell Canada v. Intra Canada Communications*.[177] In that case, Pratte J., facing a claim that anti-competitive behaviour by the plaintiff in contravention of the *Combines Investigation Act* serves as a defence in an action for copyright infringement, stated: "we entertain serious doubts that they constitute a valid defence to the action."

Clearly the Supreme Court's ruling in *Massie & Renwick Ltd. v. Underwriters' Survey Bureau Ltd.* suggests that some form of a copyright misuse defence may be available in Canada. This can be reconciled with the Federal Court of Appeals' ruling in *Bell Canada v. Intra Canada Communications*, which seemed to reject a defence of misuse, by recognizing that the former case dealt with a criminal conspiracy whereas the latter dealt solely with anti-competitive behaviour under the *Combines Investigation Act*. It has often been thought that the misuse defence, though grounded in equity, has, as its origins, principles of encouraging competition. It is not clear from these cases, however, whether a mere finding of anti-competitive behaviour will suffice in raising the misuse defence, or whether some other violation, such as a criminal act, is required. The present-day *Competition Act*, which has replaced the *Combines Investigation Act* as the guardian of encouraging competitive behaviour, now has its own statutory remedies for intellectual property misuse (discussed

---

176 *Massie & Renwick Limited v. Underwriters' Survey Bureau Limited*, [1940] S.C.R. 218.

177 *Bell Canada v. Intra Canada Communications* (1982), 70 C.P.R. (2d) 252 (F.C.A.); rev'g (1982), 62 C.P.R. (2d) 21 (F.C.T.D.).

below).[178] Whether some other form of equitable copyright misuse defence continues to exist in Canadian law remains unclear. Its existence under U.S. copyright law is much more certain.

b) Preemption of Conflicting Laws

The U.S. Court of Appeals' decision in *Lasercomb America Inc. v. Reynolds*, while clearly acknowledging the existence of a copyright misuse defence, did not deal with the specifics of reverse engineering *per se*. The sole American case to deal with the conflict between copyright and contract in the context of reverse engineering was *Vault Corp. V. Quaid Software Ltd.*.[179] In that case Heebe J. of the U.S. District Court held that contract terms which widened the scope of rights granted by the *Copyright Act*, and the legislation that sanctioned the use of these terms, were preempted by the federal copyright legislation.[180] *Vault Corp. V. Quaid Software Ltd.* concerned the validity of the *Louisiana Software License Enforcement Act*, a state statute that allowed for the enforceability of shrink-wrap licences. Heebe J. ruled that the state legislation had the effect of widening the protections granted by §106 of the U.S. *Copyright Act* and allowed contractual terms to impede the archival copy privilege conferred on authorized users by §117.[181] The Court of Appeal (5th Circuit) upheld the District Court's decision

---

178 Section 32 of the *Competition Act*, R.S.C. 1985, c. C-34, s.19 [hereinafter the *Competition Act*]. See *infra*, page 66, section entitled "Competition Law".

179 *Vault Corp. V. Quaid Software Ltd*, 655 F.Supp. 750 (E.D. La. 1987); aff'd, 847 F.2d 255 (5th Cir. 1988).

180 *Supra*, note 45, at pp. 612-3 .

181 The relevant portion of §117, that could be potentially upset by the State legislation, reads:

> Notwithstanding the provisions of section 106 [17 USCS Sect. 106], it is not an infringement for the owner of a copy of a computer program to make or authorize the making of another copy or adaptation of that computer program provided:
> ...

and held that the state legislation was preempted "because it touched upon federal copyright in a manner that set federal policy at naught."[182]

Comparatively in a Canadian context one would argue that any provincial law that expressly permits restrictions on rights and privileges guaranteed by the *Copyright Act* are *ultra vires* provincial powers. In Canada, copyright is exclusively a federal power by virtue of s. 91(23) of the *Constitution Act, 1867*.[183] This, however, does not guarantee that provinces will not create legislation in their own areas of competence that indirectly impacts on copyrighted works. The law of contract, as well as the general creation of property rights and their administration fall within the property and civil rights powers given to the provinces.[184] Accordingly, "the publication, distribution and sale of [many] forms of literature may be regulated by the province within which the publication, distribution or sale occurs. These are matters within property and civil rights in the province."[185] As a result, much of the licensing of computer programs may be said to generally fall within the competence of the provinces, although the intellectual property aspect remains a federal power. This potential for conflict, which was evidenced by the creation of the Louisiana licensing statute in the U.S., also exists in Canada.

---

(2) that such new copy or adaptation is for archival purposes only and that all archival copies are destroyed in the event that continued possession of the computer program should cease to be rightful.

182 *Supra*, note 45, at p. 612.

183 Any residual powers, not specifically enumerated by the *Constitution Act, 1867* remain with the federal government unlike the U.S. where they are left with the states. Peter W. Hogg, *Constitutional Law of Canada*, (Toronto: Carswell, 1985), at p. 86.

184 *Constitution Act, 1867*, s. 92(13).

185 *Ibid*, at p. 509.

Where such a conflict exists, the formula used to resolve the dispute in a federal system is known as the "doctrine of federal paramountcy". Under this principle, "where there are inconsistent (or conflicting) federal and provincial laws, it is the federal law which prevails. ... The doctrine of paramountcy applies where there is a federal law and a provincial law which are (1) each valid, and (2) inconsistent."[186] With legislation and, by extension, common law principles that impact upon copyright it is clear that under this same doctrine these laws may not enhance or restrict acts and privileges granted under federal copyright law.[187] This argument exists independently of any argument made by virtue of the wording of section 63 of the *Act* which expressly prohibits any rights similar to copyright from being granted other than under the *Copyright Act* (discussed below), and is concerned with both the direct and indirect impact of other legal rules upon copyrighted works and the privileges and restrictions associated with them.

c) Statutory Paramountcy

Arguably the lynch pin that most effectively secures the preemption of licensing provisions that purport to conflict with the U.S. *Copyright Act*, which ironically was not directly used in the *Vault Corp. V. Quaid Software Ltd.* decision, is section 301 of that *Copyright Act*. Section 301 states that the *Act* is the sole grantor of copyright; any

---

186 *Ibid*, at p. 364.

187 Although, "[i]t may be argued that there cannot be an impossibility of dual compliance with laws that do not impose duties, because two merely permissive laws that seem to be inconsistent can always be complied with by not doing that which is permitted." We must not look at it from the point of view of the actors involved, but from the officials who must administer the law. "If two rules would require inconsistent responses by a judge to the same set of facts then there is an impossibility of dual compliance and therefore an express contradiction." *Ibid*, at p. 256.

similar rights granted by the common law or state legislation are preempted by the *Act*.[188] By virtue of section 301(b)[189] trade secret provisions continue to apply, but only insofar as they remain consistent with the provisions of the *Act*.[190] According to one commentator,

> The far-reaching public policy Section 301 implements clearly requires preemption of contract-based protection of expression *as* expression where the effect is to secure rights in that expression which are greater than, equal to, or supplemental of those which Section 106 secures. ... The inescapable conclusion is that contractual reverse engineering prohibitions cannot survive a Section 301(a) challenge.[191]

While § 301 of the U.S. *Copyright Act* is markedly similar to section 63 of the Canadian *Copyright Act*, its construction is slightly different when it comes to the preemption of trade secret laws. The U.S. *Act* bases itself on the proposition that once copyright exists, nothing that is inconsistent with said copyright may exist.[192] Section 63 of the Canadian *Copyright Act*, however, states that nothing in the *Act* "shall be

---

188 Where other laws directly compete with copyright by creating similar right or restricting copyright, there are provisions in the Canadian, U.S., and U.K. copyright legislation that expressly state that no other law may create a copyright or similar right. Canadian *Copyright Act*, 63; U.S. *Copyright Act*, s. 301; U.K. *Copyright, Designs and Patents Act 1988*, s. 171.

189 (b) Nothing in this title [17 USCS Sects. 101 et seq.] annuls or limits any rights or remedies under the common law or statutes of any State with respect to-

> (1) subject matter that does not come within the subject matter of copyright as specified by sections 102 and 103 [17 USCS Sects. 102 and 103], including works of authorship not fixed in any tangible medium of expression; or
>
> ...
>
> (3) activities violating legal or equitable rights that are not equivalent to any of the exclusive rights within the general scope of copyright as specified by section 106 [17 USCS Sect. 106]
>
> ...

190 *Supra*, note 45, at pp. 605-6 .

191 *Ibid*, at pp. 614 - 16 .

192 See section 301(b)(3) of the U.S. *Copyright Act*, *supra*, note 189.

construed as abrogating any right or jurisdiction to restrain a breach of trust or confidence." Trade secrets, it might be argued, are given paramountcy under Canadian copyright law even where there is a conflict. In terms of reverse engineering, under this line of reasoning a finding of the existence of a protectable trade secret may lead a Canadian court to ignore the *Copyright Act* and its possible sanctioning of such an exercise.[193] Under the U.S. *Act*, if the exercise of a trade secret claim were to conflict with the rights and privileges granted under the *Act*, then according to §301 the claim will fail.

While reverse engineering may be legally prohibitable under trade secret law, where no trade secret is shown to exist, such as in the case of shrink wrap licensing which arguably fails the trade secret test for want of a "special relationship", the case for claiming copyright paramountcy is much stronger. Unlike trade secret terms, simple contractual terms are not expressly exempted from copyright principles by section 63 of the *Copyright Act*. Furthermore, section 63 does not limit its application only to other statutory instruments and is presumably wide enough to encompass contractual rights as well. Accordingly, a public interest argument that stresses both paramountcy and copyright misuse is likely to prevail where non-trade-secret contractual terms attempt to bar rights that are otherwise allowed under the *Copyright Act*.

---

193 Subject to the arguments discussed previously. See *Supra*, note 162, and accompanying text.

3. Competition Law

Even without the copyright misuse doctrine or a U.S. section 301 preemption of legislative or contractual terms in favour of copyright laws, it is arguable that an attempt to restrict reverse engineering through contractual terms may run afoul of anti-trust laws, referred to as competition laws in Canada. Grants of intellectual property protections such as patents and copyrights are exempted from the application of competition laws because of their statutory basis,[194] however, any misuse of these rights that results in the lessening of market competition in trade may be prohibited by the courts. Section 32 of the *Competition Act*, deals with the misuse of intellectual property rights and states that,

> (1) In any case where use has been made of the exclusive rights and privileges conferred by ... a copyright ... so as to
> (a) limit unduly the facilities for ... producing, manufacturing, supplying ... or dealing in any article or commodity that may be a subject of trade or commerce,
> (b) restrain or injure, unduly, trade or commerce in relation to any such article or commodity,
> (c) prevent, limit, or lessen, unduly, the manufacture or production of any such article or commodity ..., or
> (d) prevent or lessen, unduly, competition in the ... sale or supply of any such article or commodity,
> ....
> (2) The Federal Court, on an information exhibited by the Attorney General of Canada, may, ... make one or more of the following orders,
> (a) declaring void, in whole or in part, any agreement, arrangement or licence relating to that use;
> (b) restraining any person from carrying out or exercising any or all of the terms or provisions of the agreement, arrangement or licence;
> (c) directing the grant of licences under any such ... copyright ... to such persons and on such terms and conditions as the court may deem proper ...;
> ...

---

194 Section 79 of the *Competition Act*, subject to section 32 of that *Act*, expressly exempts intellectual property rights obtained under the *Copyright Act* from being deemed "anti-competitive" and thereby subject to a prohibitive order by the Competition Tribunal.

> (c) directing that such other acts be done or omitted as the Court may deem necessary to prevent any such use.

This section gives the Federal Court broad powers to restrict the misuse of any of the intellectual property rights set out therein. If it is found that reverse engineering is permitted as an exception under the copyright regime, then any licence term that would restrict reverse engineering would arguably be a use of the privileges of copyright to "unduly limit or lessen" the production of other computer program. The section 1.1 of the *Competition Act* sets out the purpose of the *Act* as being "to maintain and encourage competition in Canada ... in order to provide consumers with competitive prices and product choices". Presumably any disruption of these competitive principles runs afoul of the *Act*. To date there have been no copyright disputes arising under the *Competition Act*, and consequently the breadth of section 32 remains unknown.

In sum, although it is a common practice in the computer software industry to prohibit reverse engineering through licensing agreements, where there is no fiduciary or similar relationship between the parties, these provisions cannot be supported by trade secret laws. If it is determined that copyright law is broad enough to allow reverse engineering, as has been the case in the U.S., then these provisions will become unenforceable for a host of reasons, including: copyright misuse, federal paramountcy, statutory paramountcy and competition laws. If a valid trade secret is found to exist, then the argument to prohibit reverse engineering remains strong and potentially only competition law principles may trump it.

4. Patents

Another intellectual property regime that is increasingly important to computer program protection is patent law. Patents are monopoly rights granted by the Federal Government to inventors.[195] These rights are limited in duration (twenty years in Canada) and are fueled by two basic policy objectives: to encourage further research and development by providing economic monopoly protection to patent holders, and to create a system of knowledge sharing whereby the public may have access to patented technologies through a system of public disclosure of patent documents. Obtaining a patent does not entitle one to specific sums of money or to any other positive act, but rather is a *right to exclude others from making, using or selling the invention* for the duration of the patent. Patents are to be differentiated from other forms of intellectual property protection such as trade-marks, copyright, industrial designs, and integrated circuit topographies as patents are limited to functional articles or processes that create a tangible product. Patents are not granted for the ideas at the core of these articles or processes but instead are granted for the physical manifestation of the ideas. Traditionally, it was thought that patents were not applicable to computer programs,[196]

---

195 Patents are governed in Canada by the *Patent Act*, R.S.C. 1985, c. P-4 [hereinafter the *Patent Act*]..

196 The patent - software restriction was borne out of court decisions interpreting the scope of the respective *Patent* Acts in both Canada and the United States. The decisions in both *Gottschalk v. Benson*, 409, U.S. 63, 175 U.S.P.Q. 672 (1972), and *Schlumberger Canada Ltd. v. Commissioner of Patents* (1981), 56 C.P.R. (2d) 204 (F.C.A.), both placed steep hurdles to patenting software. The former decision held that mathematical algorithms and formulae used in making mathematical calculations or conversions were not patentable under a similar provision in the U.S. *Patent Act*, whereas the latter decision held that calculations performed by a computer are not the proper subject matter of a patent. Raymond Trudeau, "Software Patents", (1992) 9 C.I.P.R. 234, at p. 234.

however, in recent years the Canadian and U.S. Patent Offices and the courts have allowed software patents for certain products.

The basic legal test in obtaining a patent is that the invention must possess novelty, utility and some measure of inventive step (also known as "non-obviousness"). Novelty requires that the applicant be the original inventor and that the invention be the first of its kind *anywhere in the world.* There must have been no public disclosure of the invention prior to the filing of the application (subject to a one year exception in Canada, to be discussed shortly). Utility requires that the invention or process has some useful function -- in other words, it must work. Finally, the invention must be a result of ingenuity that would not have been obvious to a person of average skill in the industry.[197] Once granted a patent, the term of protection in Canada is twenty years from the date of filing the patent application, after which the invention falls into the public domain.

Although it is widely believed that software patents are prima facie not permissible under patent legislation, both the Canadian and American patent offices grant patents for computer programs whose operation results in a real world manifestation. That is, the computer program must be characterizable as "something more than a mere algorithm ... [and cannot be] merely directed to making calculations to the presentation of an algorithm and

---

[197] The degree of inventive step does not require that the invention be a revolutionary development; it may consist of an improvement on already existing technology. The rule of thumb for this test is that the invention must elicit some reaction of marvel or amazement ("why didn't I think of that?") by others in the industry. If the patent is an addition to an existing technology then any production of the invention will have to obtain the requisite authorization from patent holders of the existing technology usually in the form of licensing agreements.

its solution".[198] This view was articulated by the Federal Court of Appeal in *Schlumberger Canada Ltd. v. Commissioner of Patents*,[199] which now stands as the authoritative Canadian decision in the field of software patents. In that case, the Court held that the use of a computer could not render patentable that which was unpatentable, and since mathematical formulae are not patentable under the Patent Act,[200] they could not become patentable merely because they are in the form of a computer program.[201] The Court did not, however, state that a computer program was, as a result of its form, unpatentable. Presumably, if a computer program embodied otherwise patentable subject matter, the program would be patentable. The Canadian Patent Office, rejecting its earlier blanket ban approach to computer program patentability, seized upon this interpretation of the Schlumberger decision. As it stands, the Patent Office allows "[p]atenting inventions pertaining to novel industrial processes, installations or equipment which incorporate computer technology".[202] If the invention relates to a physical manifestation, such as an industrial process, but the novelty is primarily based in the computer program, then the patentability will depend on drafting of the patent application.[203] "Pure software" that does not relate to something other than the making of calculations is generally not considered patentable subject matter under the Act.[204]

---

198 *Supra*, note 196, at pp. 238 - 239.

199 *Schlumberger Canada Ltd. v. Commissioner of Patents* (1981), 56 C.P.R. (2d) 204 (F.C.A.); leave to appeal refused (1982), 62 C.P.R. (2d) 261.

200 No patent may be granted for mere scientific principles or abstract theorems. *Patent Act*, s. 27(3).

201 *Supra*, note 75, at p. 6-21.

202 *Supra*, note 196, at p. 241.

203 *Ibid.* A patent application consists of three parts: abstract, specification, and drawings. The abstract presents a brief summary of the contents of the specification and is often limited to one paragraph. The specification is made up of two parts: the description and the claims. The description provides a highly technical makeup of the invention such that someone skilled in the art could create the object of the patent from this description. The claims which also form part of the specification set out the limits of monopoly protection covered by the patent. The claims are

The debate with respect to reverse engineering in the realm of patents is virtually non-existent. Patents, as mentioned, are based on the fact that once a grant of monopoly patent rights is made, detailed specifications concerning the invention are made publicly available. Any use of the invention, within the scope of the patent claims filed, will require the payment of a royalty fee to the patent holder. Patent protection and trade secret protection are therefore inconsistent since trade secret protection requires that the subject matter be kept secret from public knowledge. In fact, even the act of filing a patent application, which provides disclosure of the invention to the Patent Office, may preclude subsequent trade secret protection.[205] If a patented invention is for some reason reverse engineered, then any use of the material discovered through the reverse engineering process will still potentially be covered by the claims in the patent document.[206] Accordingly, royalties would still be payable to the patent holder if the use falls within the patent claims.

---

arguably the most important and difficult part of the patent application and require the expertise of a qualified patent agent. The tension in claim drafting occurs because one must place one's invention in the context of other inventions such that it is distinct and differentiable from other inventions, while also broadly claiming protection so as to block potential infringers from successfully inventing a similar object that would not otherwise be covered by one's claims. The drawings section of the application consists of illustrations of the features outlined in the claims potion of the specification. Not all patents will lend themselves to illustration, but where possible they must be included in the application.

204 Ibid. Recently, there have been a few cases where patents been issued for software products that seem to fit the description of "pure software". Notwithstanding these idiosyncratic cases, which are few in number, the position of the Canadian Patent Office has remained firm.

205 *Kewanee Oil Co. v. Bicron Corp.*, 416 U.S. 470 (1974), at p. 484.

206 Since the granting of a patent requires that full disclosure concerning the manufacture of the invention be provided, reverse engineering is unnecessary since any member of the public may obtain a copy of this disclosure and the specifications contained therein.

Many commentators continue to believe that, because of the functionality of computer programs, patent, and not copyright, is best equipped to provide intellectual property protection.[207] Providing copyright protection to computer programs over-protects them because it improperly treats them as literary works and not solely as functional works of technology.[208] The limitations of copyright to adequately reflect the policy goals properly associated with technological works has become clearer with computer programs and has forced governments to disregard copyright when implementing protections for other technological works such as semiconductor chips.

5. Semi-Conductor Chip Protection

In the past decade, both Canada and the United States have enacted *sui generis* legislation to protect semi-conductor chips (often referred to as microchips).[209] An examination of these statutes is helpful to a discussion of copyright and reverse engineering as it provides some insight into intellectual property legislation that has been customized to address the needs of a particular technology. The recently enacted semi-conductor legislation may be contrasted with more general catch-all intellectual property legislation, such as the copyright and patent statutes. An examination of semi-conductor legislation is also helpful

---

207 See Zhou Hau, "Securing Patent Protection for Computer Program-Related Inventions", (March, 1993) Patent World 34; Stephen A. Becker, "Drafting Patent Applications On Computer-Implemented Inventions", [Vol. 4, Spring, 1991] Harv.J.L. & Tech 237; Raymond Trudeau, "Software Patents", (1992) 9 C.I.P.R. 234; *Supra*, note 75, at p.6-2.

208 *Supra*, note 35, at p. 2021.

209 An example of a microchip would be the previously discussed RAM chip. Microchips, such as ROM chips, which contain computer programs receive two-tiered protection as the computer programs are further protected by copyright legislation. *Apple Computer Inc. V. Mackintosh Computers Ltd.*, *supra*, note 18.

because it contains specific reverse engineering provisions for a computer-related technology that is otherwise protected in a manner very similar to copyright.[210]

The protection afforded by both the American *Semiconductor Chip Protection Act* ("*SCPA*")[211] and the Canadian *Integrated Circuit Topography Act* ("*ICTA*"),[212] extend to the "mask work fixed in a semiconductor chip product"[213] or topography, as it is referred to under the Canadian *Act*. Mask works, or integrated circuit topographies,[214] are defined, notwithstanding the differences in language, as a series of images that represent in totality a three dimensional rendering of the chip product. Each image represents a layer of a chip that is conceptually peeled away from the chip so as to be

---

210 According to one commentator, "[t]he overprotection of software, as compared to semiconductor chips, is particularly relevant because of the great similarities between the two technologies, both in the way they are developed and the way they operate. ... [In fact,] the dividing line between hardware (such as semiconductor chips) and software is extremely fuzzy." *Supra*, note 35, at p. 2020 and at note 83.

211 *Semiconductor Chip Protection Act*, 17 U.S.C. §§ 901 - 914 (1988).

212 *Integrated Circuit Topography Act*, R.S.C. 1985, c. I-14.6 [1990, c. 37].

213 *SCPA* § 902(a)(1) (1988).

214 Section 2 of the Canadian *Integrated Circuit Topography Act* defines topography as,

> the design, however expressed, of the disposition of,
> (a) the interconnections, if any, and the elements for the making of an integrated circuit product, or
> (b) the elements, if any, and the interconnections for the making of a customization layer or layers to be added to an integrated circuit product in an intermediate form.

§902(a)(1) of the American *Semiconductor Chip Protection Act* defines mask work as,

> a series of related images, however fixed or encoded --
> (A) having or representing the predetermined, three dimensional pattern of metallic, insulating, or semiconductor material present or removed from the layers of a semiconductor chip product; and
> (B) in which series the relation of the images to one another is that each image has the pattern of the surface of one form of the semiconductor chip product.

represented in two-dimensions. These images are protected by the respective *Acts*. In manufacturing a semiconductor chip, the mask, or topography, is superimposed onto a silicon dioxide wafer following which the mask is exposed to an ultraviolet light which has the effect of tracing the mask pattern on the wafer.[215] The traced stencil provides the piping through which electrical impulses will flow through the semiconductor chip.[216]

Both *Acts* protect the mask works or topographies of original works for a period of ten years.[217] As with the respective *Copyright Acts*, the protection afforded by the *SCPA* and *ICTA* only extends to the expression of the mask work or topography but does not protect any underlying "idea, procedure, process, system [or] method of operation" embodied in the work.[218] Although the *SCPA* requires that the mask work be fixed in order to receive protection,[219] the *ICTA* has no such requirement. The standard of originality, under both *Acts*, falls somewhere between the standard of originality[220] under the *Copyright Act*, and the standard of novelty[221] under the *Patent Act*.[222] Originality for the purposes of semiconductor chip protection under the *ICTA* requires that in addition to the work not being a "mere reproduction of another topography", it must also be the "result of an intellectual effort and ...

---

215 David Victor, "An Analysis of an Affirmative Defense for Reverse Engineering Within a System of Legal Protection for Computer Software", (1993) 66 Southern California L.R. 1705, at p. 1717.

216 *Ibid*, at pp. 1717 - 18.

217 *SCPA* §904 (a) - (b), and *ICTA* s.5 (a) - (b).

218 *SCPA* §902(c). *ICTA* section 3(3).

219 *SCPA* §901(3) and §902(a)(1). *Supra*, note 215, at p. 1718.

220 See *supra*, p. 34.

221 See *supra*, p. 69.

222 *Supra*, note 215, at p. 1718.

not ... commonplace among creators of topographies or manufacturers of integrated circuit products".[223]

The reverse engineering provisions in question also appear in similar fashion in both *Acts*. It is the appearance of these provisions that make these *Acts* distinct from the copyright provisions that apply to computer programs. Section 6(2)(a) of the *ICTA* allows for a person, to do any act "in relation to that registered topography for the sole purpose of analysis or evaluation or of research" but that does not "commercially exploit the topography or any substantial part thereof". Any act that does not copy the whole or a substantial part of the topography and that is commercially exploitive is allowed under the *ICTA*. This level of similarity is to be much more loosely construed than are similar enquiries as to copying under the *Copyright Act*. According to the proposal that resulted in the implementation of the *ICTA*,

> The reverse-engineered chip and the protected chip might legitimately be identical in electronic function and external fit. But, reverse engineering would produce a chip with a three-dimensional layout neither identical nor virtually identical to the topography embodied in the protected chip. The proposed reverse-engineering measure would legitimate the creation of a substantially similar topography for a fully compatible chip, potentially offering for example:
>
> - an improved signal/noise ratio;
> - fewer fabrication steps;
> - greater thermal stability;
> - decreased die size;
> - faster performance; and
> - lower manufacturing cost.[224]

---

223 *ICTA*, section 4(2). Similar protection is afforded by §902 of the *SCPA*.

224 Consumer & Corporate Affairs Canada - Department of Communications, "Semiconductor Chip Protection in Canada: Proposals for Legislation", Government of Canada, Department of Supply & Services, April, 1987. This report specifically recommended that, "Canada's chip-protection law should contain a reverse-engineering exception allowing the unauthorized copying of a protected topography in a process of analysis and redesign leading to the creation of a substantially similar chip topography." (At p. 49).

Infringement of a topography would result only where there the reverse engineering results in the "production and marketing: of an exact copy of a protected topograhpy; or of a chip embodying a topography virtually identical to the protected topography or to a substantial portion thereof."[225] The provisions were adopted in favour of extending copyright protection to semiconductor chips because the policy underlying chip protection recognized that "chip-protection legislation must ... suppress chip piracy without creating unnecessary obstacles to a free market in semiconductor chips and to the spread of chip technology".[226] The use of "pure copyright principles to prevent the unauthorized copying of a chip topography would not meet the needs of the semiconductor industry ... who wish to make an unauthorized copy of all of their competitor's topography for analysis; and to manufacture a substantially similar chip derived from their competitor's topography."[227] The test for valid reverse engineering under both the *ICTA* and the *SCPA*,

> suggests a two-step inquiry. First, if it is determined that a competitor has substantially studied and analyzed a protected mask work to produce its own chip, i.e. valid reverse engineering, that chip does not infringe even if it is substantially similar to the mask owner's. However, if the competitor's design incorporates identical parts of the protected design, infringement may yet be found.[228]

As with the *SCPA*, the drafters of the *ICTA* expected that the work involved in reverse engineering a chip would be substantial and would justify reverse engineering as an

---

225 *Ibid*, at p. 46.
226 *Ibid*, at p. 45.
227 *Ibid*, at pp. 43 - 44.
228 Stephen P. Kasch, "The Semiconductor Chip Protection Act: Past, Present and Future", (1992) 7 High Technology L.J. 72, at p. 77.

alternative to continually "re-inventing the wheel".[229] The concept of using a "paper trail" as proof that analysis has been performed on a chip which is allowed as a fair use under section 6(2) of the *ICTA* is native to both the Canadian and American legislation.[230] The showing that "sweat of the brow" has been expended through the use of a paper trail may involve the use of "ordinary business documents and technical materials [such as] invoices, employment and payroll records, logic and circuit diagrams, trial layouts and computer simulations of the chip".[231]

D. Conclusion

Without further legislative guidance as to the extent of the fair dealing exception to copyright infringement, copyright law prohibits the reverse engineering of computer programs. Aside from copyright, no other intellectual property regime disallows the reverse engineering of computer programs in a blanket fashion. In fact, semiconductor chip

---

229 It is estimated that reverse engineering a semiconductor chip "requires thousands of person-hours and about a quarter of the money needed to create the original chip." *Supra*, note 224, at p. 47.

230 According to U.S. House and Senate Explanatory Memoranda, 130 Cong. Rec. 28,960,

> The end product of the reverse engineering process is not an infringement, and itself qualifies for protection under the [*SCPA*], if it is an original work, as contrasted with a substantial copy. If the resulting semiconductor chip product is not substantially identical to the original, and its design involved significant toil and investment so that it is not a mere plagiarism, it does not infringe the original chip, even if the layout of the second chip is, in substantial part, similar. ... [T]he courts are not likely, as a practical matter, to find it unduly difficult to draw the line between reverse engineering and infringement, because the additional work required to come within the privilege established by § 906(a) will ordinarily leave a "paper trail."

231 *Supra*, note 224, at p. 47. The idea is that *bona fide* reverse engineering will leave a trail as compared with a pirate who "is not able to produce a long paper trail because he has not done genuine design-development work". *Ibid.*

and patent legislation either expressly allow reverse engineering or provide adequate disclosure of the ideas and processes underlying the protected work. Trade secret laws, although prohibiting reverse engineering by those in a "special relationship" with the work's owner, allow persons not in such a relationship to freely reverse engineer a computer program without penalty. If it is determined that fair dealing does allow the reverse engineering of computer programs then attempts to enhance protection, other than through trade secret laws, may be considered anti-competitive and may potentially result in the suspension of copyright enforcement privileges.

Chapter IV. Existing Approaches to the Reverse Engineering Problem

Although the problems associated with reverse engineering computer programs have not, as of yet, appeared in a Canadian context, they have received attention in other jurisdictions, most notably the United States and the European Union ("E.U.", formerly the "European Economic Community"). The E.U.'s legislative response to reverse engineering serves as a good example of how one might legislatively respond to reverse engineering issues, whereas the American experience serves as an important indicator that court challenges based on reverse engineering claims are imminent and that a legislative response should be considered as a preemptive measure, instead of leaving the issue to the uncertainty of a judicial decision based on statutory language that was not enacted in contemplation of computer technology and the novel dilemmas it raises.[232]

---

[232] The question as to the appropriate response of law to technology is a complex and profound issue and is well beyond the scope of this discussion. Clearly the judiciary must be provided with sufficient flexibility to deal with new technologies that have not been addressed by the legislature. Once recognized, though, law-makers should, if required, address the issues raised by the new technology. According to one commentator,

> Given the novelty of these phenomena, some 'social experimentation' will have to go on before we legislate. Along with the existing intellectual property rights, contract law is the privileged field for such experimentation. We should study the ways in which information is captured in contracts, both for commercial users and in mass markets.
> The law can contribute to this process by sanctioning deliberate flouting of contractually defined rights by third parties. ... Legislation in this area should follow *practice*, not the other way around. Ejan Mackaay, "Informational Goods: property of a mirage", (1985) 1 Computer Law and Practice 193, at p. 197.

A. *United States*

The issue of reverse engineering computer programs under the U.S. Copyright Act was recently considered in two U.S. Court of Appeals cases (Federal and Ninth Circuits).[233] The judgements in these cases were released a mere month apart, and together present a consistent, albeit surprising, policy oriented approach to reverse engineering. The cases mark the first time any court has dealt with the reverse engineering of computer programs under copyright laws. Prior to the release of these cases, however, the U.S. District Court was presented with an opportunity to examine the reverse engineering issue with respect to data tables. Although the former cases present more sophisticated analyses of the copyright issues involved due to their more recent release, the latter mentioned case provides insight into a rudimentary approach to reverse engineering that, according to this author, arrives at the correct conclusion.[234] The reasoning in this case reflects a certain elegant simplicity which speaks well to the issue of reverse engineering which is easily obscured by virtue of the technical nature of the subject matter being considered. A discussion of each of these three U.S. cases is set-out below.

1. E.F. Johnson Co. v. Uniden Corp. of America

In 1985, the U.S. District Court faced a situation which involved the reverse engineering of *data tables* contained in a microchip, and its permissibility under

---

A similar rationale was employed in the drafting of the reverse engineering exceptions contained in the *ICTA*. Parliament based the exception upon practice within the industry and legislated accordingly. See also the *supra*, note 227, and accompanying text.

233 *Supra*, note 49.

234 See *infra*, Chapter V. D. "Developing a Solution to the Problem", at p. 131, for a discussion of recommendations designed to deal with the problem of reverse engineering.

the *Copyright Act*. In *E.F. Johnson Co. v. Uniden Corp. of America*[235] the plaintiff moved for a preliminary injunction restraining the defendant from publishing, selling, marketing or in any way distributing software that would allow users to access a range of frequencies in the defendant's two-way radio product. The radios in question were controlled by software provided to users by the plaintiff along with the radios. The defendant reverse engineered the plaintiff's computer program, uncovering data tables that listed activation codes for the radios and then produced their own program using identical data tables. The defendant's tables, however, also contained unnecessary elements such as errors and duplications made in the plaintiff's table. The Court granted an injunction based on the substantial likelihood of success on the merits of the plaintiff's copyright claim. The Court decided that there were alternative means to copying the plaintiff's software by which the defendant could have placed the required radio codes into their own program, and that the defendants had acted in an unfair manner and were guilty of copying copyrighted expression.[236] Although this case involved data tables as opposed to reverse engineering computer instructions, the issues raised and analysis performed by the Court in construing the whether copyright expression had been used in the plaintiff's product would have been similar had the reverse engineering occurred with a computer program.

---

235 *E.F. Johnson Co. v. Uniden Corp. of America*, 623 F.Supp. 1485 (D.C. Minn. 1985).

236 *Ibid*, at p. 1502. "The Barker word was of necessity identical in both codes, but the identity of Barker word correlation techniques and sampling rates was not. [Further, the] defendant's duplication of the EFJ sample error table was not a requisite to compatibility".

The Court in this case, however, did not discuss or declare that the actions of the defendant in reverse engineering the plaintiff's computer program were infringing. Rather, the Court chose to look at the final product and examine it for traces of infringing expression. Whether intermediate copying occurred during the process of reverse engineering the plaintiff's data tables therefore remained untouched. Whether the Court's silence indicates conscious implicit acceptance of reverse engineering as a permissible act under copyright legislation remains unclear. It is more likely, that as a result of this being an early case involving computer programs under U.S. copyright law and the relatively low sophistication of the jurisprudence and analysis of the courts in this field, the *Uniden* court simply missed the issue. One may, however, discern a sub-conscious acceptance of reverse engineering implicit in the judgment through an analysis of passages in the judgment that state that had the defendants only used those functional parts of the data tables, which could not be produced through some alternate means, there would be no infringement.[237] The Court's decision presupposes that the defendant could uncover these codes which can only be done through reverse engineering the chip containing the data tables. As no statement was made to the contrary, presumably the intermediate copies of the table produced at this stage would not constitute infringing copies.

2. Atari Games Corp. v. Nintendo of America Inc.

The first of the two decisions to be released concerning the reverse engineering of computer programs, as opposed to mere data tables, was that of the Federal

---

[237] *Ibid*, at p. 1504.

Circuit Court of Appeals in *Atari Games Corp. v. Nintendo of America Inc.*. In that case, Nintendo had sued Atari for copyright infringement of its 10NES lock-out program.[238] 10NES is a program, embedded in chips contained in the popular Nintendo NES game console, which intercepts the flow of information from game cartridges once inserted into the NES console. 10NES waits for the cartridge to send a coded message[239] to the console. Once the coded message is detected, 10NES allows the console to operate the cartridge. If the appropriate message is not detected, 10NES will not "unlock" the console, and will not allow the cartridge to operate. Nintendo allowed other software developers to become licensees and write software for the NES. The licence agreements were highly restrictive but would allow the licensees to produce software which would be bundled with Nintendo's secret unlocking message.[240]

In 1986, Atari had attempted to crack the 10NES program by both monitoring the data flow between the console and the cartridges and by chemically peeling microchip layers from the 10NES chips and microscopically examining the data paths.[241] Unable to crack the 10NES program, Atari decided to become a Nintendo licensee in 1987

---

238 Nintendo had originally sued Atari for unfair competition, patent infringement, copyright infringement and trade secret violations. Atari counter-sued for unfair competition, *Sherman Act* violations, and patent infringement. The cases were consolidated and Nintendo obtained a preliminary injunction from the District Court for the Northern District of California based on Atari's unauthorized use of Nintendo's copyrighted expression. Atari appealed this injunction to the U.S. Court of Appeals, Federal Circuit.

239 The coded message, or "key", is contained in microchip form and is referred to as the 10NES slave chip. The portion of the 10NES program contained in the console is also in microchip form and is referred to as the 10NES master chip.

240 In fact, under the licensing agreements, software developers would provide Nintendo with the computer programs. Nintendo would then package the games into cartridge form with the 10NES slave chip and resell them to the software developer.

241 *Supra*, note 144, at p. 836. For a discussion of "chip peeling", see *infra*, note 269.

and was barred under the terms of that agreement from gaining access to the 10NES code and to uncovering its operation. In 1988, a lawyer representing Atari approached the U.S. Copyright Office and applied for a copy of the 10NES program which had been filed for copyright registration purposes. The lawyer stated that the code was required for litigation that had been commenced against Atari. The code was provided under the condition that "the requested copy [would] be used only in connection with the specified litigation".[242] In fact, at the time no litigation existed between the parties.

Atari used the code provided by the Copyright Office to develop its own unlocking program, entitled "Rabbit". Rabbit was a program that, although written in a different computer language, replicated the data stream provided by the 10NES slave chips. Nintendo sued Atari, and was granted a preliminary injunction for copyright infringement.

Atari's defence to the copyright infringement claim was primarily based on the fair use exception.[243] Unlike Canadian fair dealing, the U.S. doctrine of fair use has an extensive juridical history and is considerably more developed. The doctrine of fair use existed

---

242 *Ibid*, at p. 836.

243 Fair use, like its Canadian fair dealing counterpart, is an exception to copyright infringement and is defined in the U.S. *Copyright Act* as,

> the fair use of a copyrighted work, including such use by reproduction in copies or phonorecords or by any other means specified by that section, for purposes such as criticism, comment, news reporting, teaching (including multiple copies for classroom use), scholarship, or research, is not an infringement of copyright.

in the common law until its codification in 1976.[244] §107 of the U.S. *Copyright Act* sets out four factors to consider in a case where fair use is claimed:[245]

> In determining whether the use made of a work in any particular case is a fair use the factors to be considered shall include--
> (1) the purpose and character of the use, including whether such use is of a commercial nature or is for nonprofit educational purposes;
> (2) the nature of the copyrighted work;
> (3) the amount and substantiality of the portion used in relation to the copyrighted work as a whole; and
> (4) the effect of the use upon the potential market for or value of the copyrighted work.

The *Atari* court did not perform a detailed analysis of these factors in its decision. Instead, the Court looked at copyright from the perspective of balancing the interest of individual authors, who wish the rewards for their works, and society's interest in promoting the free flow of ideas. With respect to reverse engineering, the Court was not impressed with the fact that Nintendo's code was in object code form and stored on a microchip. The Court held that it is fair use to reverse engineer a computer program in order to gain a better understanding of the program's underlying ideas. On this point, the Court firmly stated,

> An author cannot acquire patent-like protection by putting an idea, process, or method of operation in an unintelligible format and

---

244 According to the Court in *Lewis Galoob Toys Inc. v. Nintendo of America Inc.*, 20 U.S.P.Q.2d 1662 (D.C. N.D. Ca. 1991),

> [the copyright] statute was 'merely intended to restate the present judicial doctrine of fair use, not to change, narrow or enlarge it in any way.' H.R. Rep. No.. 94-1476, *supra*, at 5680. Congress had 'no disposition to freeze the doctrine in the statute, especially during a period of rapid technological change.' *Id.* (At note 5, p. 1668).

245 The four factors are "non-exclusive 'factors to be considered' in assessing fair use; they are intended to guide but not to limit analysis". *Ibid*, at p. 1668.

> asserting copyright infringement against those who try to understand that idea, process, or method of operation. ... The Copyright Act permits an individual in rightful possession of a copy of a work to undertake necessary efforts to understand the work's ideas , processes and methods of operation.
>
> This permission appears in the fair use exception to copyright exclusivity.[246]

Fair use protects the intermediate stages of reverse engineering and the related copies that are produced as a result, from being deemed as infringing. Fair use, according to the Court, however, did not extend to reverse engineering with the purpose of profiting by replicating protected expression. Fair use only applies where reverse engineering is necessary to discern unprotectable ideas.

In the case of Nintendo's 10NES program, the Court determined that the underlying unprotectable ideas were those codes that are required to unlock the NES console. Anything in excess of that was presumed to consist of protectable expression.[247] Atari's copy of various parts of the 10NES program included errors and deletions made by Nintendo in their program. This copying of unnecessary portions of the Nintendo table resulted in a decision against Atari for copyright infringement since, as the Court stated, "[t]hese unnecessary instructions strongly suggest that the Rabbit program is substantially similar to the 10NES program ... [which] Nintendo is likely to show ... contains protectable expression. ... Nintendo is likely to prove substantial similarity between the Rabbit and 10NES programs sufficient to support its infringement claims".[248] Furthermore, Atari's unfair appropriation of

---

246 *Supra*, note 144, at p. 842.
247 *Ibid*, at pp. 843 - 44.
248 *Ibid*, at p. 845.

the 10NES source code by misrepresenting its intentions to the Copyright Office pre-empted any use of the fair use defence since fair use only applies to authorized copies of a work.[249] Atari's dishonest behaviour in obtaining the 10NES code also precluded Atari from using a copyright misuse defence, based on Nintendo's extensive licensing agreement, to preclude Nintendo's assertion of its copyright.[250] The Court also failed to recognize the possibility that Atari copied elements unnecessary to unlock the Nintendo console in order to ensure against "the possibility that Nintendo could alter its console in the future to utilize currently unused portions of the compatibility code."[251]

Although Atari lost on its appeal to lift the injunction imposed by the District Court, the loss was due both to the dishonest way in which Atari obtained the code from the Copyright Office and to the fact that Atari copied portions of Nintendo's code in wholesale fashion. The Court did, however, rule that the copying required to reverse engineer a program, both in terms of intermediate copies and the final derivative work product, are a fair use if the copying is necessary to "understand the ideas and processes in a copyrighted work." According to the Court, the purpose of the reverse engineering is central to the defence of fair use. If the purpose of reverse engineering is "to profit from replicating protected expression" then fair use cannot be invoked. Furthermore, "[a]ny reproduction of protectable expression

---

249 *Ibid*, at p. 843; and *Harper & Row Publishers v. Nation Enterprises*, 471 U.S. 539 at pp. 562 - 563, 105 S.Ct. 2218 at p. 2232.

250 *Ibid*, at p. 846. See *supra*, note 170, and accompanying text.

251 Harold C. Moore, "Atari v. Nintendo: Super Mario Uses "Expressive" Security Feature to "Lock" Out the Competition", (1992) 18 Rutgers Computer & Technology L.J. 919, at p. 933. This effect manifested itself in the *Sega Enterprises v. Accolade Inc.*, 24 U.S.P.Q.2d 1561 (9th Cir. 1992) decision. See *infra*, note 254, and accompanying text.

must be strictly necessary to ascertain the bounds of protected information within the work".[252]

3. Sega Enterprises v. Accolade Inc.

Immediately following the Federal Circuit Court of Appeals' decision in *Atari Games Corp. v. Nintendo of America Inc.*, the Ninth Circuit Court of Appeals released its decision in *Sega Enterprises v. Accolade Inc.*,[253] another case involving the reverse engineering of home video game software. In that case, Accolade had sought to reverse engineer cartridges containing video game programs with the intention of gaining an understanding of how the software interacted with Sega's Genesis game consoles. Using the results of the reverse engineering, Accolade created a development manual for use by its programmers in developing games that would operate on the Sega console. The information used by Accolade at this early stage was purely functional and did not involve any literal copying of Sega program code. During the period that Accolade was beginning to release its own compatible software, Sega, in order to combat high degrees of piracy of its games in South-East Asia, decided to licence a copy protection system and place it in its newly developed Genesis III consoles. The trademark security system ("TMSS") would be placed in each Sega console and would, when a cartridge was plugged in, search for the letters "SEGA" contained in the program code of each cartridge. Upon

---

252 *Supra*, note 144, at p. 843. Ideas, expression that is merged with the idea, and expression that is not fixed are not protected by copyright. See *supra*, Chapter III.A.1.a. "The Idea/Expression Dichotomy", at p.29. The fair use exception, according to the Court, would apply in cases where protectable expression is copied in the process of reverse engineering, but that such copying is necessary to understand the limits of what is and is not protectable within the work.

253 *Sega Enterprises v. Accolade Inc.*, 24 U.S.P.Q.2d 1561 (9th Cir. 1992).

finding this initialization code, the Genesis III console would produce a display of "PRODUCED BY OR UNDER LICENCE FROM SEGA ENTERPRISES LTD.", and would unlock the console for game play much like Nintendo's 10NES system.

Upon finding that its game cartridges would not work with the Genesis III system, Accolade examined the results of its reverse engineering efforts and found that each licensed Sega cartridge contained a small unused portion of code in each game's "power-up" sequence which was now being used by Sega in its TMSS lockout system.[254] Accolade created a small header file (20 - 25 bytes of data), which contained the SEGA initialization code, to be used in all future game development and also incorporated it into its development manual. The effect of this header file was to allow Accolade's games to work on Genesis III consoles. However, because of the Sega TMSS program, the Accolade cartridges would also cause the display of Sega's start-up trademark message.

Sega brought an action against Accolade for copyright infringement, trademark infringement and false designation of origin. Accolade countered with claims of false designation of origin, unfair competition, false or misleading statements and

---

254 The use of apparently useless code in an upgrade of the system highlights a difficulty inherent in the idea/expression analysis performed by courts in order to determine what may fairly be reverse engineered and used in a competing program. If computer code, and/or errors, serve no purpose then, as was the case in both the *Atari Games Corp. v. Nintendo of America Inc.* and *E.F. Johnson Co. v. Uniden Corp. of America* decisions, they may not legally be copied. However, the original program developers may subsequently use these pieces of computer code in a lockout system which would render all other manufacturers' products unworkable. This possibility was ignored by the court in the *Atari Games Corp. v. Nintendo of America Inc.* case. See *Supra*, note 251, and accompanying text.

intentional interference with prospective economic advantage. Sega was granted a preliminary injunction by the U.S. District Court based on Sega's copyright and trademark infringement claims.[255] Accolade appealed to the U.S. Court of Appeals (Ninth Circuit), whereupon the injunction was lifted and the matter was remanded to the District Court.

The decision of the Court of Appeals in lifting the preliminary injunction provides a detailed analysis of the reverse engineering issue as it applies to computer programs. The Court's finding, as with the *Atari* court, that reverse engineering is allowed is based on the fair use exception under the *Copyright Act*. Unlike the *Atari* court's decision, the *Sega* court provided a step-by-step analysis of the four fair use factors, and a discussion of three related reverse engineering arguments: intermediate copying; the idea/expression dichotomy as a possible justification for reverse engineering; and a section 117 defence. Although Accolade ultimately succeeded in its claim that its reverse engineering constituted a fair use, the Court rejected Accolade's other three arguments.

With respect to intermediate copying, the Court held that the fact that the product is intermediate, as opposed to final, in nature in no way exempts it from the application of the *Copyright Act*. Whether intermediate copying properly constitutes an infringement of copyright must be examined by the courts, notwithstanding whether the

---

255 *Supra*, note 34.

final product itself constitutes an infringement.[256] This, the Court reasoned, is based on the fact that,

> the computer file generated by the disassembly program, the printouts of the disassembled code, and the computer files containing Accolade's modifications of the code that were generated during the reverse engineering process all ... [fall] squarely within the category of acts that are prohibited by the statute.[257]

In terms of Accolade's claim that the idea/expression dichotomy provides a justification for reverse engineering, the Court ruled that the fact that ideas may be contained within protected expression does not mean that the expression that surrounds those ideas may be ignored. Accolade had argued that the nature of computer programs make them different from other more traditional copyrightable works whose ideas may readily be perceived and understood by human beings. Consequently, Accolade argued that reverse engineering was simply a crutch to put computer programs on an equal footing with these other works.

The final argument disposed of by the Court in Sega's favour concerned the claim that section 117[258] of the *Copyright Act* provides that the intermediate copies made during the reverse engineering of a computer program is not an infringement of the *Act*. The Court reviewed the CONTU report responsible for the implementation of section 117,[259] and ruled that,

> Accolade's use went far beyond that contemplated by the CONTU and authorized by section 117. Section 117 does not purport to protect

---

256 *Supra*, note 253, at p. 1566.
257 *Ibid.*
258 See *Supra*, note 115.
259 *Supra*, note 42.

> a user who disassembles object code, converts it from assembly into source code, and makes printouts and photocopies of the refined source code version.[260]

The Court did, however, accept Accolade's fair use argument and ruled that,

> in the case before us, disassembly is the only means of gaining access to those unprotected aspects of the program, and because Accolade has a legitimate interest in gaining such access (in order to determine how to make its cartridges compatible with the Genesis console)), we agree with Accolade. Where there is good reason for studying or examining the unprotected aspects of a copyrighted computer program, disassembly for purposes of such study or examination constitutes a fair use.[261]

a) The Sega Court's Fair Use Analysis

Of the four factors mentioned in §107 of the U.S. *Copyright Act*,[262] the Court found in favour of Accolade on the first, second and fourth factors, while finding in favour of Sega on the third factor.

The first factor concerns the purpose and character of the use including whether such use is commercial in nature or for nonprofit educational purposes. The Court ruled that "[t]he commercial nature of a use is a matter of degree, not an absolute" and in the case of Accolade the direct purpose of reverse engineering the Sega code was to ensure the compatibility of their cartridges with the Sega console.[263] The commercial

---

260 *Supra*, note 253, at p. 1568.
261 *Ibid.*
262 For a list of the factors, see *supra*, note 245, and accompanying text.
263 *Supra*, note 253, at p. 1569.

nature of cartridges was secondary and was, at best, indirect. In considering this factor, the Court also seemed to hint at the a "sweat of the brow" rationale combined with some measure of creativity.[264] That is, where the reverse engineering involves some creative effort, the court will look favourably upon a defense of fair use. In this regard the Court held that "there is no evidence in the record that Accolade sought to avoid performing its own creative work ... it wrote its own procedures. ... [T]hese facts indicate that ... its direct use of the copyrighted material, was simply to study the functional requirements for Genesis compatibility."[265] In support of its holding on this point, the Court also noted that notwithstanding the potential commercial gain to Accolade, the public interest in promoting compatibility, and hence competition, will reduce the strength of Sega's claim that Accolade stands to gain commercially.

With respect to the fourth factor listed in §107, the effect of the use upon the potential market for or value of the copyrighted work, the court held that Accolade's development will not directly cause Sega to lose customers since the games developed by Accolade are not copies of those developed by Sega.[266] Although the court recognized that Accolade's entering the market will "undoubtedly affect" the market for Genesis

---

264 The application of "sweat of the brow" to copyright protection has recently gained much notoriety as a result of the U.S. Supreme Court's decision in *Feist Publications Inc. v. Rural Telephone Service Inc.*, 499 U.S. 340 (1991). This case raised the issue of whether or not copyright could exist in a telephone book. The U.S. Supreme Court held that mere "sweat of the brow" was not sufficient in itself to attract copyright protection. Creative effort, including the particular selection or arrangement, of the data was also necessary for copyright to exist.

265 *Ibid*, at p. 1570.

266 The Sega Court's decision seems to indicate that in order to successfully argue this factor against fair use, it must be shown that "the new work ... supplant[s] the direct market for the particular copied work". David L. Hayes, "The Legality of Disassembly of Computer Programs", (1993) 12 Computer/Law Journal 1, at p. 8.

games, albeit indirectly, the public policy rationale underlying copyright legislation is to promote creative expression and not to stifle competition in a specific market. Accordingly, notwithstanding "the minor economic loss Sega may suffer", the court decided this point in Accolade's favour.[267]

The final factor decided in Accolade's favour was the second one listed in §107, the nature of the copyrighted work. The basis of the Court's analysis of this factor was that computer programs are essentially utilitarian in nature and they contain many elements whose expression is often dictated by notions of efficiency, or external factors such as compatibility and other industry demands.[268] These elements are not protectable under copyright legislation. Accordingly, if Accolade was unsuccessful in its fair use claim, Sega would obtain a monopoly over the functional ideas underlying its work which is not supposed to fall within the ambit of copyright protection. Sega argued that there were alternative methods, other than disassembly, of uncovering its initialization code such as chip peeling[269] and clean room procedures[270]. The Court disagreed with Sega and

---

267 *Supra*, note 253, at p. 1571.

268 *Ibid.*

269 This procedure was also used by Atari in an unsuccessful attempt to reverse engineer Nintendo's 10NES program (see *Supra*, note 241, and accompanying text). "Chip peeling" refers to a process whereby layers of microchips are microscopically dissected in order that their data pathways may be mapped out. From these maps one may reconstruct the object code contained in the chips. According to the *Sega* court, relying on the *Nintendo* decision, "chip peeling yields only a physical diagram of the *object code* embedded in a ROM chip. It does not obviate the need to translate object into source code." *Ibid*, at p. 1572.

270 A "clean room" procedure is a measure taken by computer program developers that attempts to ensure that programmers do not copy existing competing programs. Programmers are isolated in that they are only provided with the desired functional specifications, and are not given access to the competing product. Independent development of similar programs is not considered to be an infringement of copyright law. *Ibid.*

found that "disassembly of the object code in Sega's video game cartridges was necessary in order to understand the functional requirements for Genesis compatibility".[271] In effect, the Court recognized that while Accolade's disassembly did necessarily involve intermediate copying which was a potentially infringing act, to allow Sega to succeed on this claim would expand its copyright beyond its intended scope and stifle competition. Accordingly the Court was faced with choosing the lesser of two evils and based on public policy reasons found that Accolade's actions constituted a fair use of Sega's copyrighted material. The Court did not clarify whether this balance would have indeed tipped in Sega's favour had alternative methods to disassembly been available to Accolade.

With respect to the third factor mentioned in §107, the amount and substantiality of the portion used in relation to the copyrighted work as a whole, the Court found in favour of Sega, although this finding was not sufficient to tip the fair use argument in favour of Sega. Accolade disassembled entire programs written by Sega in order to uncover the initialization sequence and there could be no argument that the amount of code copied was anything less than 100% of the whole. The *Sega* court, however, quoting from the decision in *Sony Corp. v. Universal Studios*,[272] ruled that "[t]he fact that an entire work was copied does not ... preclude a finding of fair use"[273] The Court felt that since the purpose of copying the entire work was incidental to its

---

271 *Supra*, note 253, at p. 1572.
272 *Sony Corp. v. Universal Studios*, 464 U.S.417; 220 U.S.P.Q. 665 (1984).

desired use which was not directed at copying the work, the fact that entire work was copied was "of very little weight."[274] The Court's ruling on this point runs counter to that of the Canadian Exchequer Court in *Zamacois v. Douville*[275] which ruled that where an entire work is copied the fair dealing defense under the Canadian *Copyright Act* cannot be invoked. *Zamacois v. Douville* continues to be relied upon as being the authoritative case defining the scope of the fair dealing defence under Canadian law. Clearly this case was decided in an era where copyright did not contemplate the protection of computer program works whose functional nature may dictate a revision of those principles applied to traditional literary works. This "special functional nature" of computer program works was clearly recognized by the *Sega* court in its analysis of the idea/expression dichotomy.

B. *The European Union*

The European Union, formerly the European Economic Community, was formed by the *Treaty Establishing the European Economic Community*[276] in 1958. The purpose of this Treaty was to form a common economic market by encouraging the free flow of goods and services through the elimination of trade barriers between member-states.[277] The *EEC Treaty* was further expanded by the *Treaty on European Union*.[278]

---

273 *Sony Corp. v. Universal Studios*, 464 U.S. 417 (1984), at pp. 449 - 50; *Hustler Magazine, Inc. v. Moral Majority, Inc.*, 796 F.2d 1148 (9th Cir. 1986), at p. 1155 ("the copying of an entire work does not preclude fair use *per se*.").

274 *Supra*, note 253, at p. 1573.

275 See *Supra*, note 126 and accompanying text.

276 The *Treaty Establishing the European Economic Community* (*EEC* Treaty) is also commonly referred to as *The Treaty of Rome*.

277 *EEC Treaty*, Articles 30 - 34.

278 *Treaty on European Union*, February 7, 1992, Europe Documents No. 1759/60. Also known as *The Maastricht Treaty*.

The area of intellectual property, and more specifically copyright, is not provided for by these treaties and has been a difficult area to harmonize. According to one commentator, "[t]here will not be a unified European copyright law in the near future. The national copyright laws of member states will continue to apply ...".[279] More urgently, however, has been a recognition by the European Commission that specific industries, including the software industry, cannot wait for a general harmonized community copyright law. Because of their relative economic importance and the disparity of treatment among various member-states, these industries require that common legal rules be applicable throughout member-states without delay.

1. The E.E.C. Directive on Computer Programs

Accordingly, in order to harmonize existing protections and establish a level of common principles with respect to the intellectual property of computer programs within its various member-states, the European Council passed the *Directive on*

---

279 Silke von Lewinski, "Copyright in the European Communities: The Proposed Harmonization Measures, (1992) 18 Brooklyn Journal of International Law 703, at p. 703. In an attempt to strengthen copyright law within the Community, in 1990 the European Commission adopted a proposal that would require accession, by all member states, to various international copyright and neighbouring rights treaties including: the *Berne Convention for the Protection of Literary and Artistic Works*, September 9, 1886, revised July 24, 1971, 828 U.N.T.S. 221 ; and the *Rome Convention for the Protection of Performers, Producers of Phonograms and Broadcasting Organizations*, October 26, 1961, 496 U.N.T.S. 43. (At p. 708).

the legal protection of computer programs[280] in 1991.[281] Fundamentally the objectives of the Directive[282] were two-fold:

> to prevent the unlawful copying of computer software, or "computer piracy", within the Community by ensuring an adequate level of protection for those who create computer software; and
>
> to promote the free circulation of computer software within the Community and allow industry to take advantage of the single market by harmonising the national laws of the Member States relating to the use and reproduction of computer software.[283]

The European Commission, charged with designing the Directive, opted for copyright as the basis for protection because of its flexibility in interpretation, its implicit and continual balance of monopoly rights with societal interests, its limitation of protection to expression and not ideas, and most importantly because to-date copyright had been the computer program - intellectual property protection of choice of a number of Community member states.[284] The Commission's choice of copyright was hardly revolutionary or controversial. However, the decompilation[285] exception to infringement was hotly debated prior to its inclusion in the Directive.

---

280 *Directive on the legal protection of computer programs* (91/250/EEC) [1991] OJ L122/42 [hereinafter "the Directive].

281 At the time of passage, E.E.C. membership included twelve member-states: Belgium, Denmark, France, Germany, Greece, Ireland, Italy, Luxembourg, the Netherlands, Portugal, Spain, and the U.K..

282 A Directive is not enforceable in itself, but directs member-states to incorporate its provisions into national laws. It is not required that the incorporation of these provisions be verbatim but rather that the effect of the Directive be captured. The implementation of Directives is time limited by provisions contained within the Directive itself, although extensions may be granted to member-states which require more time in order to comply with the provisions. Where the national legislation does not adequately reflect the spirit of the Directive's provisions, the Commission is responsible for filing suit against the member-state before the European Court of Justice. *Supra*, note 102, at p. 9.

283 *Supra*, note 102, at p. 5.

284 *Ibid*, at pp. 5 - 6.

285 The Directive employs the term "decompilation" in a broad sense, equivalent to the term "reverse engineering", as opposed to its literal meaning. "Decompilation" as used in the

Early in its evolution, the Commission's Proposal for the Directive (the "1989 Proposal") contained two sections which could be interpreted as potentially allowing the reverse engineering of computer programs protected under the terms of the Directive. Section 1.3 of the 1989 Proposal stated that legal protection was to apply only to a computer program's expression, and not

> to the ideas, principles, logic, algorithms or programming languages underlying the program. Where the specification of interfaces constitutes ideas and principles which underlie the program, those ideas and principles are not copyrightable subject matter. ... [286]

The second relevant clause contained in the 1989 Proposal was Article 5.1 which allowed the reproduction and adaptation of a computer program, potentially including the reverse engineering thereof, where such reproduction and adaptation was necessary for the use of the computer program. This adaptation right would exit notwithstanding a refusal by the copyright-holder. Although implicitly the gist of these sections would allow reverse engineering, the 1989 Proposal did not contain express wording to that effect. The European parliament accepted the Commission's 1989 Proposal subject to certain amendments. The Commission responded with its amended proposal in 1990 which contained express wording that would allow reverse engineering in certain circumstances. Article 5*bis* of the Commission's 1990 proposal was eventually adopted almost verbatim

---

Directive therefore includes disassembly as well. See *supra*, note 36, and accompanying text; and *supra*, note 37, and accompanying text.

286 Section 1.3 of the European Commission's 1989 Proposal as quoted in: J.D. Byrne, "Computer programs and reverse engineering: Recent European Developments", (1991) 8 Canadian Computer Law Reporter 45, at p. 47.

into the Directive (as Article 6), and is unique in that it expressly allows reverse engineering in the context of copyright protection of computer programs.[287]

Article 6(1) states that decompilation may be used where it is "indispensable to obtain the information necessary to achieve the interoperability of an independently created computer program with other programs". Article 6 goes on to list three conditions which limit the scope of the decompilation, all of which must be met, in order for this copyright exemption to apply. First, the individual(s) decompiling the software must be licensees or others with a right to use the software; second, the information being sought through decompilation must not be already readily available; and third, the decompilation must only be carried out on those parts of the software that are necessary to gain the information being sought.

Article 6(2), in order to further clarify and strengthen the meaning of Article 6(1), emphasizes those objectives that are not allowed when decompiling a

---

[287] The Commission's 1990 proposal, as background, stated,

> ... the Commission has been persuaded that the original proposal, which left the matter of "reverse engineering" not explicitly regulated, lacks sufficient clarity. It is therefore proposed that an additional Article 5bis dealing with a derogation allowing "reverse engineering" of programs for the purposed of interoperability of the program should be added. Nothing in this Directive should prevent however the "reverse engineering" of a program, whether incorporated into hardware or not, under the conditions of Article 5bis for the purpose of independently creating an interoperable program, wherever it may be incorporated.

As found in *ibid*, at p. 48.

computer program. Anyone seeking to decompile a computer program is prohibited from using any information retrieved through the use of the decompiling exception in order to achieve something other than interoperability, or in order to undertake any other goal that violates copyright, such as developing software similar in expression.[288] Clearly, in its request that the Commission include an express reverse engineering section, the European Council felt it necessary to make its intentions, as expressed in articles 1.3 and 5 of the Commission's 1989 Proposal, as certain and unambiguous as possible due to their revolutionary and controversial stance. The Council wished to ensure that any

---

[288] The enacted Directive contains a definition of interoperability in its preamble. The Directive states:

> Whereas the parts of the program which provide for the interconnection and interaction between elements of software and hardware are generally known as "interfaces";
> Whereas this functional interconnection and interaction is generally known as "interoperability"; whereas such interoperability can be defined as the ability to exchange information and mutually to use the information which has been exchanged. ...

Critics of the Directive were quick to point out that the original proposals failed to adequately define the term "interoperability." One interest group that had pushed hard on this point was The Software Action Group for Europe ("S.A.G.E."). S.A.G.E. vehemently argued that without an adequate definition of "interoperability", decompilation could be used for purposes other than those intended by the Directive. That is, without the establishment of strict definitions of the level at which a program may validly interface with another, software developers may decompile in a manner that violates the author's protected expression.

On the other side of the coin, in support of the E.E.C. Directive's decompilation provisions are The European Committee for Interoperable Systems ("E.C.I.S."), and other similar groups, who favour a more standardized computer environment. Although all groups, including S.A.G.E. and other opponents to the decompilation provisions, favour standardization in the computer industry, the two sides prioritize their concerns differently. Groups such as S.A.G.E. feel that on a balance, there is more to lose than gain through allowing decompilation in this fashion, while groups such as the E.C.I.S. charge that although their concerns are not altogether without foundation, groups, such as S.A.G.E., are overly alarmist and unrealistic in their assessments. Sunny Handa, "The E.E.C. Software Directive and its Impact on the Development of Computer Standards", (Paper written at the Faculty of Law, University of Toronto, 1991) [unpublished], at pp. 23 - 24.

decompilation would be both limited in scope and subject to strong constraints against its abuse.

2. Implementation of the Directive's Provisions in the U.K.

The provisions of Directives do not in themselves have any enforceability until they are implemented into national legislation. Directives, however, are binding, and member states must implement them within the time prescribed by each Directive.[289] The implementation of Directives does not require that member-states adopt the exact wording used in the Directive, although, they must implement legislation to the same effect. The deadline for implementing the Software Directive was January 1, 1993, although that deadline was met by only a handful of member-states, including Denmark, Italy and the U.K..[290] Other E.U. member-states still have not formally ratified the Directive's provisions into their national laws. The procedure for dealing with member-states that fail to comply with the implementation date set out in a Directive, or with any other provision,[291] is for the Commission to bring the member-state before the European Court of Justice, whose decisions are binding on national courts. Often, however, the Commission will allow member-states to extend the deadline for implementing Directives, especially where the member-state's domestic law requires a significant change.

---

289 *Supra*, note 102, at p. 9.
290 Clifford Chance, "Computer and Communications Bulletin", March 1993, Issue 9, at p. 3.
291 *Supra*, note 102, at p. 9.

As mentioned, the U.K. has implemented the changes required by the Directive, although its interpretation of the Directive in its domestic law has strayed from the exact wording of the original text. The changes to U.K. legislation were incorporated by the Copyright (Computer Programs) Regulations 1992 which came into force on January 1, 1993.[292] The Directive's reverse engineering provisions have been incorporated into section 50B of the U.K. *Copyright, Designs, and Patent Act 1988* but they do not supplant, in any way, the *Act*'s fair dealing provisions[293] which may also be used to potentially permit reverse engineering.[294] Section 50B allows the reverse engineering of a computer program in order:

> (a) to convert it into a version expressed in a higher level language, or
> (b) incidentally, in the course of so converting the program, to copy it.

Section 50B(2) restricts the use of reverse engineered material as required by Article 6(2) of the Directive, to situations where:

> (a) it is necessary to decompile the program to obtain the information necessary to create an independent program which can be operated with the program decompiled or with another program ('the permitted objective'); and
> (b) the information so obtained is not used for any purpose other than the permitted objective.

---

292 *Supra*, note 36, at p. 591.

293 U.K. *Copyright, Designs, and Patent Act 1988*, 1988, c. 48, section 29(1). As with section 27(2) of the Canadian *Copyright Act*, fair dealing under the U.K. *Act* for the purposes of research or private study is not an infringement of copyright.

294 It has been pointed out that with the existence of an arguably broader fair dealing exception, a separate reverse engineering provision is unnecessary. Fair dealing may, in fact, allow reverse engineering for a host of reasons not contemplated by the Directive such as commercial research. *Supra*, note 36, at pp. 595 - 96.

The U.K.'s implementation of the Directive's provisions has been criticized for only exempting the reverse engineering of lower level language code into a higher language 50B(a) whereas the "Directive is more generous, allowing translation, adaptation, arrangement or alteration."[295] Another potential troublespot in the U.K. legislation is section 50B(2)(b) which potentially restricts a software developer, who has reverse engineered a computer program in order to create a compatible one, from later deciding to use the information retrieved to create another, potentially competing program.[296] The use of the 'permitted objective' mechanism seems to dictate a sense of immediacy -- that the reverse engineering must be conducted with a given objective in mind, and the results only used in conjunction with that objective. Under a strict interpretation of the *Act*'s wording, the reverse engineering may have to be reworked should the developer desire to create other compatible computer programs at some later date.

Another notable point, especially for Canadian observers, is that under Reg. 12(2) of the U.K. *Act*, the ability to reverse engineer a computer program cannot be pre-empted by any term in a licensing agreement executed after the implementation of the provisions on January 1, 1993; any attempt to do so would render such a term void.[297] As mentioned, in Canada and the U.S., the current practice is to prevent reverse engineering

---

295 *Ibid*, at p. 594. Implicitly, the U.K. *Act* would seem to prohibit the translation of a work to an equivalent level language such as from binary code to hexadecimal notation.

296 *Ibid*, at p. 595.

297 *Ibid*. This differs from the right, as found in section 50C of the *Act*, to copy or adapt a program as required for its use. The legislation makes no mention of the inability to contract out of this provision.

through licensing agreements. Even if reverse engineering is found to be lawful under the Canadian fair dealing exception, often a user will have to contend with overcoming the licensing agreement, presumably using the defences set out in Chapter III. C. 2. above.

Notwithstanding various technical differences, the U.K. *Act* has captured the general spirit of the Directive's reverse engineering exception although arguably its choice of wording serves to indicate a certain discomfort and hesitation with lifting copyright restrictions through such legislative exceptions. The U.K. legislature seems to prefer deferring the task of fine tuning copyright protection to the courts through the more general fair dealing and public interest exceptions.[298]

C. *Australia*

If Canadians were to look around the world for another legal jurisdiction that parallels itself in terms of facing computer copyright issues, that jurisdiction would undoubtedly be Australia. The Australian *Copyright Amendment Act 1984* expressly placed computer programs within the scope of copyright protection, but as with the Canadian amendments, passed in 1985, left the proposition rather vague. In terms of the evolution of case law, Canada and Australia are also similarly poised. Whereas a Canadian court recently rendered the first trial decision dealing with the "look and feel" issue,[299] and the protection of non-literal elements in a computer program, Australian

---

298 Whether this discomfort is peculiar to Anglo-American systems of copyright, as opposed to continental *droit d'auteur* regimes, is a topic worthy of further study but is beyond the scope of this paper.

299 See *Delrina Corp. v. Triolet Systems Inc.*, *supra*, note 14.

courts have not yet faced the challenge.[300] The situation with reverse engineering, however, is the reverse.

1. Autodesk Inc. V. Dyason

In 1992, the Australian High Court rendered its decision in *Autodesk Inc. V. Dyason*,[301] a case that concerned the reverse engineering of data tables found within a computer program. Once again, the subject matter of the reverse engineering process concerned a security lockout scheme employed to protect against software piracy. The difference between *Autodesk Inc. V. Dyason* and the American cases, however, was in the manner in which the reverse engineering was conducted. In this case, the defendants used an oscilloscope to map out the signals being sent between a hardware key, referred to as the "AutoCAD lock", and the computer program, known as "Widget C". In order to run the AutoCAD program, an engineering design application, the AutoCAD lock must be introduced into a computer port. When the AutoCAD program is executed it in turn runs the Widget C program which generates a stream of computer signals, based on a data look up table, which are sent the AutoCAD lock, manipulated, and sent back. Only where the returning signals match the appropriate profile, as contained in Widget C, will the AutoCAD program be permitted to proceed with its execution. The defendants used an oscilloscope to detect the signals flowing both to and from the AutoCAD lock, and figured out the "transitions" being performed by the hardware device. The defendants

---

300 Sheila McGregor, "Look & Feel - Australia", (1993) 1 Mealey's Litigation Reports 44, at p. 44.

301 *Autodesk Inc. V. Dyason* (1990), 18 I.P.R. 109 (Aust. Fed. Ct.), reversed (1992), A.I.P.C. 90,855, 22 I.P.R. 162 (Aust. H.C.).

then constructed their own "Auto Key lock" and marketed it as a substitute for the AutoCAD lock. Presumably, a large number of purchasers of the defendants product were to be persons with pirated copies of the AutoCAD program but who lacked the AutoCAD lock necessary to make the program run.

Presumably the issue of intermediate copying was not addressed by the Australian High Court in its decision because the defendants used an oscilloscope and not a disassembler. Had a disassembler been used, the Court would have been presented with an intermediate copy in the form of computer code and may have dealt with the issue. Since the disassembly was performed with an oscilloscope, thereby yielding a translation of the original data tables in a non-conventional format, the Court ignored this aspect of the copying. Instead, the Court concentrated on the final use of the uncovered expression and ruled that the defendants reproduced a substantial part of the Widget C program in their Auto Key lock and, as a result, infringed the plaintiff's copyright. The Court adopted a very strict interpretation of the language of the *Act* which holds that a computer program may be in any material form and in any notation.[302] The Court also noted that the stream of digits was not random and was therefore worthy of copyright protection.[303] The Court ignored the fact that the data reproduced was both unique and its purpose purely functional, and that there was no other way of creating a sequence that would unlock Widget C.

---

302 *Ibid*, at p. 174.
303 *Ibid*, at p. 169.

Clearly this case did not raise issues of creating independent interoperable computer programs which could work in conjunction with AutoCAD. Whether the Court, in such cases, is willing to override its finding of copyright infringement in favour of a fair dealing exception remains unclear. However, it should be noted that the Court's decision foreclosed upon at least one compatible product, the Auto Key lock. While it is arguable that fair dealing cannot apply in the creation of directly competing products, this reasoning, and consequently the scope of a fair dealing and public interest defence, was never considered by the Court and as a result leaves one with the troublesome impression that perhaps this decision was taken without a thorough appreciation of the potential compatibility issues that are often raised in reverse engineering cases.

Notwithstanding these worries, the decision is sufficiently narrow, and non-specific with respect to the process of reverse engineering, that a future Court may allow reverse engineering through a fair dealing or public interest exception without directly conflicting with it. As with the decision of the U.S. District Court in *E.F. Johnson Co. v. Uniden Corp. of America*, the Autodesk court did not oppose the existence of reverse engineering as a process. The Court simply concentrated upon the defendant's final product and, rightly or wrongly, came to a factual conclusion that it did contain protectable expression copied from the plaintiff's program. Whether ignoring the reverse engineering question implies tacit permissibility remains unclear. The decision of the U.S. District Court in *E.F. Johnson Co. v. Uniden Corp. of America*, while similarly

ignoring the reverse engineering issue, is logically structured in such a manner that a tacit acceptance of reverse engineering can more readily be inferred than in the instant case.[304]

D. Conclusion

Each of the aforementioned judicial decisions have either avoided a finding of reverse engineering and intermediate copying as an infringement of copyright or have found that copyright fair use provisions are sufficient to allow the process. None of the decisions thus far has opposed reverse engineering as an intermediate process. The decisions all properly concentrate on the final product and conduct their infringement tests at this stage. With the addition of the European Union member-states who are bound to implement reverse engineering provisions in their national legislation, there is a clear trend in allowing reverse engineering notwithstanding that it *prima facie* constitutes an infringing act. Each of the aforementioned jurisdictions has chosen to concentrate its enquiry on the ultimate use of that which was gained during the process. If the use has been to copy protected expression then a finding of infringement has been made, subject to considerations of compatibility. The determination as to whether the expression is indeed protected by copyright is based on standard principles of copyright law, such as whether a work is more properly idea or expression or whether it is only expressible in a singular manner (doctrine of merger).[305] If the work is protectable then, as was the case in *Sega Enterprises v. Accolade Inc.*,[306] the court may look to issues of compatibility and hold that a defense of fair use applies. In the case of *Sega Enterprises v. Accolade Inc.*,

---

304 See *supra*, note 237, and accompanying text.
305 See *supra*, note 81.

the Court applied the ultimate finding of fair use to both the final product and to the intermediate copying performed by the defendants in reverse engineering the plaintiff's product. The following Chapter attempts to further reconcile these different approaches and provide an analysis of reverse engineering in the light of the goals of copyright law, after which several solutions designed to deal with the problem of reverse engineering in a Canadian context are recommended.

---

306 *Supra*, note 253.

Chapter V. Justifying the Reverse Engineering of Computer Programs

As mentioned, conceptually, reverse engineering moves from a finished product towards its underlying ideas. Copyright law, we have seen, only protects the expression of ideas but not the ideas themselves. Furthermore, the reverse engineering of a computer program, whose expression is protected by copyright law, is likely prohibited since the process used in reverse engineering will produce intermediate copies of the computer program which runs afoul of copyright laws. Even where the goal of the reverse engineering is to obtain the unprotectable ideas, the process of creating intermediate copies must be undertaken, and will be considered an infringing act. This unique and novel conundrum facing copyright law begs the question whether copyright law should be modified to create an exception to the problem of intermediate copying, and in what situations should such an exception be applied. Clearly these questions must be viewed in the context of copyright policy and its intended goals.

Answering the former question, whether an exception should indeed be created, is a complex issue and requires investigation of the fundamental principles underlying copyright law. Accordingly, the following sections present a rudimentary analysis of copyright from an economic perspective which seems to present the basic rationale for the existence of copyright protection. Once the former question is dealt with, the latter question, concerning the scope of the reverse engineering right, is much simpler to answer.

A. *The Economics of Copyright Law*

The study of economics and its place in law has received a great deal of attention in the past three decades. This resurgence of interest, dating back to 1960, best termed the "new" law and economics, concerns the application of economic "theories and empirical methods" to areas that were not traditionally thought of as being capable of such analysis, such as judicial decision making.[307] Traditionally, the economic analysis of law concerned areas related to antitrust and economic regulation, where the language of economics was already in wide use. Whereas the former economic analysis concerns the explanation of laws and judicial decisions that regulate non-market behaviour, the latter is concerned with government intervention in explicit markets.[308]

The study of intellectual property laws, and specifically copyright policy, arguably has its foot in both camps. Intellectual property protections, including copyright, seek to impose monopoly rights in market economies where competition is perceived as a normative good. These diametrically opposed goals co-exist for the simple reason that intellectual property protections seek to repair what is perceived as an anomaly in an otherwise competitive economic system. This "anomaly" is a result of the unique "public good" characteristics of intellectual property that are not found in traditional "tangible" property, and will be further discussed in Section V. B. below. Intellectual property protections, it is felt, use anti-competitive rules as a means to achieving a more

---

307 Richard A. Posner, "The Economic Approach to Law", (1975) 53 Texas L.R. 757, at p. 759.

308 Richard A. Posner, "Some Uses and Abuses of Economics in Law", (1979) 46 The University of Chicago L.R. 281, at pp. 281 - 282.

competitive end.[309] In this way, an economic analysis of copyright can be said to fall into the traditional branch of law and economics.[310]

In another sense, however, the development of copyright law through judicial decision making, as well as the behaviour of parties towards one another, notwithstanding the rules of copyright, can be said to more properly fall within the domain of the "new" approach to law and economics. As a result of this duality, the discussion of reverse engineering that follows, will sway between both of these defined areas. After all, the demarcation between them, if at all relevant, is an academic one and is tenuous at best.

Another important feature of the law and economics movement that has both received a great deal of external criticism and created internal turmoil is the distinction between the normative and positive approaches to law. The normative approach seeks to define "what should be", whereas the positivist merely seeks to explain behaviour using economics as its fundamental underpinning but does not put forth a position as to whether economic goals are indeed worthy of pursuit.[311] The reason for the divergence of these approaches is rooted in the fact that the criticisms leveled at those

---

309 For an excellent discussion of the economics of copyright law, see W.M. Landes and R.A. Posner, "An Economic Analysis of Copyright Law", (1989) 18 The Journal of Legal Studies 325.

310 *Supra*, note 308, at p. 282.

311 *Ibid*, at p. 285.

engaged in the normative approach[312] cannot be satisfactorily countered. After all, the moral position that economic efficiency and wealth maximization ought to be pursued is highly subjective and highly debatable. The positive school of law and economics, recently supported by the works of Posner and Coase, seek to evade these charges by claiming their works are merely explanatory and predictive.[313]

The economic analysis of copyright law from a normative viewpoint is, however, less debatable since the goal of copyright law is to balance, economically, the rights of society against those of individual authors: "striking the correct balance between access [to a work] and incentives [in creating a work] is the central problem in copyright law".[314] The argument that cultural protection and protection of the integrity of the authors is the central feature of copyright law may be more true in *droit d'auteur* jurisdictions, such as France, but given the history of Anglo-American copyright, and the limited moral rights protections found therein,[315] this argument does not hold true in Canada or other jurisdictions boasting a similar approach to copyright.

---

312 For example, G. Calabresi, "Some Thoughts on Risk Distribution and the Law of Torts", (1961) 70 Yale L.J. 499.

313 Whether one can realistically avoid making a normative claim is the subject of much debate. Simply beginning with certain behavioural assumptions, as does the positivist school, implies some normativity. See *infra*, note 316, and accompanying text.

314 *Supra*, note 43, at p. 326. According to Posner, "[s]tudies of regulated behaviour, although often strictly positive in content and purpose, have an important role in the formulation of policy and thus contribute to the normative economic analysis of law". *Supra*, note 308, at p. 286.

315 See *Supra*, note 77. The U.S. *Copyright Act* does not protect moral rights of copyrighted works, except works of visual art (§106 of that *Act* protects the integrity and attribution of these works). The United States continues to object to both Articles 6^{bis} and 18 of the Berne Convention which provide for moral rights and the protection of works existing at the convention's entry into force, respectively. Accordingly, the U.S. has not implemented these sections into its domestic law. § 104 of the U.S. *Copyright Act* prevents direct reliance of the Convention's provisions and states that,

In sum, broadly speaking, the economic approach to the study of law is concerned with how laws and legal decision-making promotes efficiency and wealth maximization. The economic approach assumes that people are rational economic actors and will generally gravitate towards these goals.[316] As an example, legal rules may maximize wealth by increasing production yield, by optimally equating price and quantity or by reducing transaction costs between parties.

> [n]o right or interest in a work eligible for protection under this title may be claimed by virtue of, or in reliance upon, the provisions of the Berne Convention, or the adherence of the United States thereto. Any rights in a work eligible for protection under this title that derive from this title, other Federal or State statutes, or the common law, shall not be expanded or reduced by virtue of, or in reliance upon, the provisions of the Berne Convention, or the adherence of the United States thereto.

The U.S. also is currently exempt from having to grant moral rights protection to foreign authors. Annex 1701.3 of the North American Free Trade Agreement further excludes the United States from any moral rights obligations that arise under Article 6^{bis} of the *Berne Convention* under that Agreement. In the U.K., sections 79 and 81 of the *Copyright, Designs and Patents Act, 1988*, c. 48, state that moral rights -- including the right of an author to be identified in conjunction with the work, as well as the right to the integrity of the work -- do not apply in the case of computer programs. Moral rights were only brought into the Canadian *Copyright Act* in 1988 as part of *An Act to amend the Copyright Act and to amend other Acts in consequence thereof*, S.C. 1988, c. 15 [now R.S.C. 1985 (4th Supp.), c.10]. See section 14.1, 14.2, 28.1 and 28.2 of the *Copyright Act*. No specific moral rights exemption applicable to computer programs exists in the Canadian *Act*.

316 *Supra*, note 307, at p. 761. "Economics is the science of rational human behaviour". *Supra*, note 308, at p. 287. According to one commentator, the positive study of law and economics assumes "that the rules, taken as a whole tend to look as though they were chosen with a view to maximizing social wealth (economic output as measured by price)". Frank Michelman, "A Comment on Some Uses and Abuses of Economics in Law", (1979) 46 The University of Chicago L.R. 307, at p. 308.

B. *The Economics of Intellectual Property Rights*

The subject matter of intellectual property protection is often described in the language of economic theory as "public goods." Public goods are those goods which are characterized by non-exclusivity and non-rivalry. The concept of non-exclusivity denotes the feature that a good, once produced, is equally available to all members of a group (i.e. society) irrespective of their contribution in producing the good.[317] The concept of non-rivalry occurs where the use of a good by one person will not affect its use by others.[318]

The basic model for setting the optimal, or efficient, price of a traditional, non-public, good in a competitive market is to pinpoint the price at the point where it is equal to the cost of producing the last unit (marginal cost).[319] With a public good, this does not work as the marginal cost of producing the good is theoretically zero (or very close to zero). Accordingly, it is expected that with a public good, such as a computer program, competitors will flood the market with copies of a work thereby forcing the price of the work towards zero.[320] If the price goes below the author's cost of producing the work, then, given that s/he is a rational economic actor, the work will not be produced.[321] In order to counter this effect, some form of monopoly protection, in the

---

317 David Schmidtz, "Contracts and Public Goods", (1987) 10 Harvard Journal of Law & Public Policy 475, at p. 475.

318 *Ibid.*

319 R. Lipsey *et al.*, *Economics 5th ed.*, (Harper & Row: New York, 1982), at p. 200.

320 *Supra*, note 43, at p. 328.

321 The cost of producing a copyrightable work can best be thought of as comprising two components: the cost to the author of producing the work, and the cost of copying and distributing the work. In order for a work to be created, the difference between expected

form of legal rules, is required in order to combat the effects of non-rivalry and non-exclusivity thereby giving the work characteristics naturally attributable to non-public goods.[322]

It is well documented that in order to promote efficiency with tangible property, as opposed to intangible property such as intellectual property, certain pre-conditions are necessary:

> (i) *Universality*: all scarce resources should be owned by someone.
> (ii) *Exclusivity*: property rights should be exclusive rights.
> (iii) *Transferability*: this is necessary to ensure that resources will be transferred from low-valued uses to high-valued uses.[323]

This list forms the basis for granting property rights protection in western society, and will consist of additional factors depending on who one asks. However, the list provided is often considered to contain the core requirements for creating a regime of property rights, and is generally agreeable to all.[324] Because of the aforementioned features of non-exclusivity and non-rivalry, absent any legal rules to the contrary, public goods do not generally display any of these characteristics.

---

revenues and the cost of copying and distributing the work must be greater than the cost to the author of creating the work. *Ibid*, at pp. 326 - 27.

322 Other examples of public goods include: national defense, police protection, road construction, and environmental protection. *Supra*, note 317, at pp. 475 - 76.

323 Frank H. Stephen, *The Economics of the Law*, (Ames: Iowa State University Press, 1988), at p. 14.

324 Other features identified might include: durability -- that property rights must be granted for substantial periods and cannot be merely transitory; that inaccessible and unique resources are not made the object of property rights protections; and that generally individuals, as opposed to groups, should be given property rights. Ejan Mackaay, "Informational Goods: property of a mirage", (1985) 1 Computer Law and Practice 193, at p. 195.

If a legal regime were to impart these rights to public goods, such as computer programs, then the other economic concepts that generally underlie modern government policy towards markets can be more readily applied to these works since they would now display similar characteristics with other traditional forms of property and can therefore be more readily managed.[325] Unfortunately the story does not end there, since by granting owners of "public good" works these characteristics, the law is guaranteeing the owners of the works extremely high returns for their efforts since these owners may continue to duplicate and distribute their works at virtually no cost forever. Accordingly, in order to regulate the rate of return, the law provides a time limit on intellectual property works[326] after which the works fall into the public domain and assume the characteristics of public goods.[327]

---

325 According to Prof. Mackaay, "[i]nformation[al goods are] a peculiar commodity. Traditional commodities are captured in law as physical goods. But information does not coincide with physical support. ... This poses particular problems to lawyers: the law traditionally attaches itself to material forms; yet the content, the information, which is immaterial, eludes it." *Ibid*, at p. 194.

326 Copyright is protected in Canada for the life of the author plus fifty years (*Copyright Act*, section 6). There has been much debate as to whether the length of time that computer programs are protected should be lessened since, given the rate of technological progress, after fifty years computer programs will be obsolete. Furthermore, as a result of the long term of protection, and the purchasing characteristics associated with computer programs, copyright holders stand to make super-normal returns. In 1984, facing a revision that eventually expressly placed computer programs under the jurisdiction of the *Copyright Act* in Canada, a proposal was presented to the Canadian government that would have limited the length of protection of computer programs under the *Act* to a five-year term (see *From Guttenberg to Telidon: A White Paper on Copyright*, Consumer and Corporate Affairs, Government of Canada, Ottawa (1984), at Section XII). This proposal was not acted upon and currently the term of protection for computer programs under the *Copyright Act* is the same as for any other literary work (see *A Charter of Rights for Creators*, Sub-Committee on the Revision of Copyright, House of Commons, Government of Canada, 1985, at Recommendation 60). This lengthy term of protection has remained consistent with the approach taken by other jurisdictions.

327 Roger E. Meiners & Robert J. Staff, "Patents, Copyrights, and Trademarks: Property or Monopoly", (1990) 13 Harvard Journal of Law & Public Policy 911, at p. 913. Also see, *supra*, note 324, at p. 195.

Thus far the analysis provided has presented a rudimentary way of looking at the basis of creating intellectual property rights.[328] Once these rights are created, the task of fine tuning them to suit the broader policy goals of optimally encouraging research and the dissemination of information may be undertaken in a relatively controlled environment that remains consistent with instruments of government policy in other sectors of the economy.[329]

C. *The Economics of Reverse Engineering*

Framed in the language of the instant debate, the economic analysis of whether to permit reverse engineering under the law of copyright must examine the benefits of this activity in light of the goals of copyright law. Once again, copyright law balances authors' rights with societal access with a view to optimally maximizing societal wealth[330] through the creation of monopoly rights in an environment that broadly seeks to encourage competitive behaviour.[331] [332]

---

328 The pricing of intellectual property, however, is not generally restricted by compulsory licensing provisions, with few exceptions, and accordingly economic theory would dictate that a monopolist will produce a quantity of goods where marginal cost is equal to marginal revenue as opposed to price. The result of this behaviour is to set price at a point greater than the marginal cost of producing the good. Accordingly, the quantity produced is restricted and the price is increased as compared with a competitive market. *Supra*, note 319, Chapter 13. The problems associated with the monopoly pricing of intellectual property goods were recently made apparent in the Canadian pharmaceutical industry where the compulsory licensing patent scheme was terminated in favour of a traditional patent/monopoly regime. The public outcry that came as a result of this government action concerned public fears that drug prices would rise. A monopolist may also engage in price discrimination in order to capture more of the consumers' surplus This latter form of behaviour, however, is more likely to be regulated by consumer protection and business practices legislation.

329 Roger E. Meiners & Robert J. Staff, "Patents, Copyrights, and Trademarks: Property or Monopoly", (1990) 13 Harvard Journal of Law & Public Policy 911, at p. 912.

330 Copyright is not primarily concerned with individual rights *per se*. Rights attributed to individuals are merely incidental to the broader purpose of maximizing societal wealth. This is evidenced by the wording of Article I, Section 8, of the U.S. Constitution which states that Congress has the power "[t]o promote the progress of science and useful arts, by securing for

1. Software Compatibility

The benefits of reverse engineering are obtained through the creation of compatible programs and a standardized environment which in turn "offers rewards by making programs easier to use, providing greater productivity, and offering greater networking capabilities."[333] Societal wealth, in the context of technological progress, is maximized by facilitating the creation of a greater number of computer programs, which is a function of the remuneration received by authors.[334] If authors are adequately

---

limited times to authors and inventors the exclusive right to their respective writings and discoveries." Clearly, the protection individual rights is recognized as being necessary to achieve the goal of "societal progress." If copying were freely allowed, the incentive for authors to create works would diminish to the detriment of society at large. In such a situation, costs of obtaining works would also increase since "[t]here would be a shift toward the production of works that are difficult to copy; authors would be more likely to circulate their works privately rather than widely, to lessen the risk of copying; and contractual restrictions on copying would multiply." *Supra*, note 43, at p. 332.

331 A typical example of imposing monopoly protection in order to increase efficiency and hence societal wealth is the local telephone service providers. Unlike intellectual property, local telephone systems because of their structure are often referred to as "natural monopolies". Allowing many local telephone service providers would reduplicate expenditures and create massive economic efficiencies since users would be on different networks. Furthermore, as with the case of reverse engineering computer programs, one of the key benefits of telephone monopolies appears in the form of standardizing access to the system to newly deregulated long distance services providers, a process which is currently in its infancy in Canada.

332 According to Prof. Mackaay,

> [m]uch of western society is premised on the widest possible availability of information. From the proposition that information should be as widely available as possible it may be concluded that information should circulate freely. ... [We] accomplish this by creating property rights in it, whose main feature is precisely the power to of the owner to exclude everyone else from it. ... [Much] as we want certain commodities to be widely available, their creation may require special efforts and these efforts will only be forthcoming if rewards are promised. Property rights are a means to provide such rewards.

Supra, note 324, at p. 194.

333 *Supra*, note 35, at p. 2030.

334 Western economies seems to thrive on and encourage technological progress, notwithstanding whether or not it is a normative good. Technological progress is generally regarded as a societal benefit. See *supra*, note 330. When we talk of societal wealth maximization, in the context of

protected, and hence remunerated for their efforts, they will continue to produce programs. If the scope of copyright protection is too high, the remuneration payable to authors will be too high, and will cause a fewer number of computer programs to be disseminated through society, and in terms of creating a prohibition to reverse engineering, will stifle the creation of compatible/interoperable programs.[335]

A typical countervailing argument, that leaving a ban on reverse engineering would only redistribute wealth in favour of the original program developer, as opposed to reducing societal wealth, rings false. If a computer programmer has to pay a licensing fee to link his/her program to another program, this feature may be left out as the fee may fatally impact on the programmer's cost/benefit analysis of writing the computer program. Reducing the number of compatible programs, would negatively impact on societal wealth for several reasons. Without the ability to transfer data between programs, users are more likely to stay with one program, notwithstanding whether it is best designed to perform the user's tasks. The cost of switching the data would be too high, and the use of inefficiently written programs would become pervasive. With an interoperable program, that understands another program's data format, the user has a

---

reverse engineering, we are really talking about reducing inefficiencies and the wastage of scarce and valuable resources which may be put to better use. Whether these resources are indeed better used in creating a greater number of products which is equated with fueling technological progress is a purely personal question that cannot be answered objectively.

335 Encouraging "[c]ompatibility allows innovative new products to enter a monopolized market and lowers development costs by allowing a programmer to attach his single innovative component to a preexisting complete system", in effect not forcing the programmer to reinvent the wheel. *Ibid.*
Put another way, "[t]he less extensive copyright protection is, the more an author, composer, or other creator can borrow from previous works without infringing copyright, and the lower, therefore, the costs of creating a new work". *Supra*, note 43, at p. 332.

clear choice between which program better suits his/her needs without incurring a significant cost in switching between the two.

2. The Costs/Benefits of a Standardized Computing Environment

With respect to the creation of a standardized computing environment, whether the creation of such a standardized environment is indeed an activity that should be encouraged is best examined by framing the issue in economic terms. Arguably the most important benefit to creating standards is that it saves scarce resources by not forcing computer programmers to continually re-invent the wheel each time they write a new computer program. Instead, a standardized piece of code that is widely available allows programmers to use their creative energies and other resources to build upon an existing standardized base. Although many software developers have their own internal standards for use by "in-house" programmers, clearly this wealth maximizing effect will be more pronounced should the standards exist on an industry-wide basis. Furthermore, scarce resources will be re-directed to further fueling technological progress, which although debatable, seems to be viewed generally as a normative good.[336]

Related to these savings, is the fact that as development costs decrease, more programmers will be able to afford to enter the market thereby increasing competition, which remains consistent with the spirit of society's economic structure and

---

[336] See *supra*, note 334.

is again generally considered to be a normative good.[337] Even if these new market players do not possess an absolute advantage over existing firms, they will possess a comparative advantage through specialization, thereby allowing a more efficient allocation of resources resulting in higher levels of output and lower prices.[338] A grant of monopoly protection to any one market player will create barriers to entry and increase trading friction amongst market participants through licensing costs, thereby opposing this effect.

From the consumer's point of view, this standardized environment appears beneficial since prices for goods may be brought down as development costs are saved through the use of established standards. Aside from the actual product price savings, consumers of software also stand to benefit from savings generated through less training and retraining time being required to learn new software packages. For example, in terms of creating a standardized user interface environment,[339] society benefits by not having to retrain individuals to operate entirely different programs with different command sequences and screen layouts.[340]

---

337 As competition increases, market imperfections are reduced and the pricing of a product becomes more efficient in terms of resource allocation and quantity produced. A perfectly competitive market benefits all parties involved by offering the lowest price to consumers and the maximum revenues to the industry.

338 *Supra*, note 319, at pp. 350 - 51.

339 Reverse engineering may apply to the creation of standardized user interface environment in cases where add-on programs that operate in conjunction with the original are sought. A simple example might be a dictionary designed to work in conjunction with a word processor created by someone other than the word processor's creators. Ideally the dictionary could be activated with a command sequence and instantaneously read words that are highlighted, or blocked off, within the word processor. In this way, the operation of the user interface of the word processor must be understood by the dictionary's programmers.

340 The costs to business end users who have been forced to switch computer software can be extremely high. These costs include the cost of the software as well as the cost required to re-train individual workers which may include: courses and any related lost time at work, lost productivity as familiarity

Notwithstanding the aforementioned benefits, the creation of a standardized environment is not altogether without its difficulties. The prime disadvantage is arguably the risk of "going down the wrong path" of technology, and only after having spent a great deal of resources, find out that it is wrong. In such a case, the transaction costs required to "get on the right track" may be too great and accordingly the correct path may never be chosen since to start over may, at that point, be too costly an endeavour.[341] A less dramatic form of this effect is simply the risk of choosing a less efficient technological standard and getting stuck with it, thereby incurring opportunity costs relative to the better technology. Unfortunately, this risk is not quantifiable, and consequently it is difficult to compare it to the associated advantage of creating a standardized environment. Exactly why a particular standard is chosen depends on a host of factors including, but not limited to, initial cost, availability, ease of use, flexibility, marketing, and so forth. The mix of factors will vary with each standard and is generally also not predictable.

Two recent examples of the negative effects associated with the adoption of a standard, stand out: the rejection of the Betamax™ video tape standard in favour of

---

with the product increases, as well as errors attributable to the switch in software. It has been estimated that the cost of re-training a secretary on a word processor without standardized commands can run as high as $1000 U.S.. *Supra*, note 30, at p. 20.

341 Based on hindsight, "[o]nce an industry has been standardized, it can be extremely difficult to break out of that standard, even if it is no longer optimal." *Supra*, note 35, at p. 2028.

the VHS standard, and, more related to the instant discussion, the development of the Personal Computer architecture by IBM in the 1970s.

With the former example, the Betamax standard, originally patented and obtainable via license from Sony Corp., was clearly the technologically and functionally superior videotape format relative to the competing VHS standard primarily developed by JVC Corp.. However, because the VHS standard was an "open" standard, useable by all, it gained widespread acceptance over the Betamax format which was difficult and expensive to licence. As a result, Betamax video tapes and recorders have largely faded from existence.

A second and more relevant example to the topic at hand was the adoption of the PC architecture for personal computers developed in the late 1970s. At the time, computer engineers did not foresee the imminent explosion in computer processing and computer memory technology which came about as a result of rapid technological progress and dropping prices for computer chips. Accordingly, the architecture for the original IBM PC, which has now pervaded our society through the distribution of PC "clones", limited its memory accessing abilities to 640 Kilobytes (KB).[342] This restriction is now widely referred to in the computer industry as the

---

342 This restriction was originally based upon the processors used (Intel 8088 processor) which could only address a maximum of 1 megabyte (1024 KB) of memory. The engineers at IBM, feeling that 640KB was sufficient for application programs, reserved the upper 384KB of address space for special functions such as video memory, and the built in basic programming language. Over time this architecture became accepted by hardware and software manufacturers, and developed into an industry standard. Even though today's computer programs have

"640KB barrier". Generations of PC technology computers that followed on the heels of the hugely successful PC were also forced to incorporate the 640KB barrier in order to retain compatibility with other generations of personal computers. To many manufacturers and users, it was determined that retaining compatibility with the PC standard was a priority even though alternate, more effective, technologies were readily available soon after the 640KB limitation became a reality. Some of the reasons for remaining loyal to this inferior standard were based on cost, marketing, and availability of compatible software. Anyone in the computer industry is acutely aware that the limitations brought about by the 640KB barrier have hampered software development over the years and have thereby resulted in huge inefficiencies reflected as opportunity costs (economic losses) in the computer software industry, and society as a whole.

Related to the "down the wrong path" disadvantage is the possibility that where a standard becomes pervasive, this will create a certain technological myopia in the industry. That is, once a standard is adopted it may forestall or at least hinder new technological approaches to problems. The extreme of this thinking, for the purposes of clarity, holds that every problem would be formulated only in terms of existing technology, thus precluding creative research into alternative solutions to the problem. Once again, however, the disadvantages will not always exist in reality. Whether other better solutions even exist, based on the available technology, in each given instance is

---

expanded their memory requirements well beyond the 640 KB limit, the memory space between 640 KB and 1024 KB (1 megabyte) must continually be treated as reserved under the PC architecture. Although this has resulted in inefficiency and consternation on the part of

not known, and the amount to which this myopia deters research from other potentially valuable areas is also uncertain.

Notwithstanding whether or not the development of a standardized environment is a benefit or hindrance, absent any intervention to the contrary, the development of standards in the computer industry appears to be a naturally occurring phenomenon. One reason for this natural occurrence is that users often need to share computer data files, generated by various application programs, with one another. One example might be an accounting firm that does the books for various businesses. This firm will recommend that its clients use specific software for in-house recordkeeping, such as point-of-sale software, that is compatible with the firm's own accounting, or report generating, software. As a result, the client may simply provide its accountant with its data files at the end of each fiscal period. Once these types of relationships develop, an industry standard begins to develop, since users want to retain the greatest level of compatibility to give themselves a greater selection of potential persons they can communicate with, thereby reducing their costs (through greater competition). Newly created small businesses, for example, will attempt to purchase that software being used by other small businesses and by accountants doing small business work in order to obtain a greater choice of accountants which in turn will be available at a cheaper price.[343]

---

computer hardware and software developers, the barrier continues to exist with no purpose other than to maintain the standardized environment that has developed.

343 As the number of participants increases in a market, the more perfect the market becomes. See *supra*, note 337, and accompanying text.

Software developers themselves are often the greatest proponents of a standardized environment. One of the most effective ways of marketing a computer program is to make it compatible with as many existing programs as possible. The newer program, presumably incorporating new technological advantages, will allow users of existing programs to easily switch to the new program. Once the software developer has captured the market it seeks, and develops a standard of its own, it often changes its tune and demands that its standards be protected in order that it may exercise monopoly power over its users thereby obtaining a greater level of profits. Two often cited computer industry examples of this effect are the Apple Macintosh™, and Lotus 1-2-3™ user interface standards. In both cases these companies based their own products on existing technology. In the case of Apple, their iconic interface coupled with the use of a mouse and pointer on the screen was first developed by the Xerox Palo Alto Research Center and the Stanford Research Institute in the 1960s and 1970s,[344] whereas Lotus 1-2-3,™ originally written in 1982, was based on the Visicalc™ spreadsheet originally developed by Daniel Bricklin, a Harvard Business School student, for the Apple II computer in the late 1970's.[345]

---

344 *Supra*, note 19, at p. 330.

345 Although Lotus Corp. sued others for copying its spreadsheet design (see, for example, *Lotus Development Corporation v. Paperback Software International and Stephenson Software Ltd.*, 740 F.Supp. 37 (D. Mass. 1990)) and *Lotus Development Corporation v. Borland, Inc.*, 788 F.Supp. 78 (D. Mass. 1992)), "so too did Lotus borrow heavily from *Visicalc* - and in particular from *Visicalc*, Advanced Version, for the Apple III". "Taking the Stand: The Look and Feel Issue Examined", *PC Magazine*, May 26, 1987.

Another example of a natural move towards adopting standards has been the increase use of object oriented programming. Object-oriented programming consists of using standardized routines, or mini programs, known as objects, and tying them together with both other objects and with one's own code in order to create a finished product. Objects are generally fine tuned to accomplish a specific task and are generally superior to similar program procedures that are created each time a program is written. The advantages of using an object-oriented approach allows more effective programs to be written more efficiently.

In sum, a standardized computing environment has both economic benefits and costs associated with it, although the costs tend to be more uncertain than the benefits, which are more readily quantifiable. If one were to choose whether or not to pursue such a standardized environment based on this knowledge, one would choose to pursue the goal. Furthermore, since the phenomenon is naturally occurring, pursuing the goal involves little or no legal intervention. To prevent the development of standards would require legal intervention in order to counteract the naturally occurring forces, and would also result in defeating those benefits associated with a standardized environment set out above. The underlying justification required to pursue this latter course of action should involve something more than merely unquantifiable probabilities of negative consequences occurring. Mere uncertainties are not sufficient to prompt intervention into this naturally occurring economic order. The proverb, "a bird in hand is better than two in the bush" would seem to apply itself to this thinking. Accordingly, the pursuit of a

standardized computing environment, based on available information, should, at best, be considered an economic good, and, at the very least, merit a *laissez-faire* approach.

If we accept the proposition that a standard environment is a deserving economic goal for society as a whole, then to grant monopoly protection over such standards would reduce its potential efficiencies,[346] and require a strong reason for doing so. An analysis of copyright law, whose fundamental goal is to optimally protect authors so as to encourage the creation of works in order to maximize the goal of societal progress in the arts and/or technology, does not yield any clear signals as to an increase in this optimality should reverse engineering be prohibited. At worst the signals are mixed, and at best they lean in favour of allowing reverse engineering since it leads to the development of a standardized computing environment. To place an artificial hurdle in the way of standardization based on unknown factors and effects is a dangerous proposition. If reverse engineering is to be prohibited, more convincing data and a clearer indication of the prohibition's beneficial effects must be demonstrated. Furthermore, the consequences of, in effect, granting monopoly protection over whatever standards do develop requires convincing justification in light of the goals of copyright law set out above. As it stands, granting monopoly protection over standards is a far cry from the objectives of copyright legislation.

---

346 Providing monopoly protection over standards that develop would, in fact, counter the positive effects of standards creation as was exemplified by demise of the Sony Betamax videorecorder standard.

D. Developing a Solution to the Problem

Once it is accepted that reverse engineering is a behaviour that should be encouraged, a solution that works within the larger framework of copyright protection must be developed. The two models to choose from are: creating a statutory exception (the European approach), or expanding the fair dealing exception to cover reverse engineering (the U.S. approach). Both methods have their own advantages and comparative disadvantages.

1. Creating a Statutory Exception

A statutory exception provides certainty, and can be used to clearly delineate the desired scope of the exception. Furthermore, the use of a statutory exception would allow the legislature to expressly preclude overrides of these copyright terms through licensing and trade secrets law as is the case with the U.K. legislation.[347] Presently, the use of licensing to override copyright terms is a common practice although where the two directly conflict, the courts may be able to rule of the copyright terms based on public policy. In cases where a valid trade secret relationship is found, pre-empting such overrides using an expansion of the already existing fair dealing exception is even more difficult, since the Canadian *Copyright Act* does not give the courts the jurisdiction to override trade secrets in support of copyright principles. It is likely that under the present copyright regime, the use of a valid trade secrets argument would be sufficient to supplant an argument of fair dealing under the *Copyright Act*. As mentioned,

---

[347] See *supra*, note 297, and accompanying text.

however, it is unlikely that the mere use of shrink wrap licensing will constitute a sufficient relationship between the parties so as to allow the trade secrets argument to be successfully invoked.

Because the use of trade secrets in cases involving reverse engineering is not altogether consistent with the breach of confidence pre-emption clause in section 63 of the *Copyright Act*,[348] a statutory section drafted with the purpose of allowing reverse engineering should correct for this uncertainty. Accordingly, such a section would expressly provide that, in the case of reverse engineering computer programs, the breach of confidence exception outlined in section 63 of the *Copyright Act* shall not apply.[349] Furthermore, since we cannot foresee future challenges to current thinking, unlike section 50B of the U.K. *Act*, we should draft the statutory section without being overly inflexible, and avoid, as best we can, any mention of specific technologies. We should allow a broad reverse engineering right of an authorized (legitimate) copy of a computer program. The section should expressly provide that it is an exception to intermediate copying only for the purposes of reverse engineering and not for use of the materials once the reverse engineering is complete. If at the end of the reverse engineering process, those reverse engineering the product wish to use the fruits of their labour, they will still have to

---

348 There exists a strong argument that the fundamental purpose in using a trade secrets argument to prohibit reverse engineering is to prevent competition, as opposed to restraining a breach of confidence. See *supra*, note 162, and accompanying text.

349 Implementation of such a section will undoubtedly require a redrafting of section 63 so that its concepts, of exclusive jurisdiction of the *Copyright Act* and of an exception for breach of confidence laws, can be delineated into sub-parts, such as 63(a) and 63(b). If, for example, the breach of confidence exception was placed in 63(b), the reverse engineering exception would then read "section 63(b) shall not apply to the operation of this section".

respect the limits of the idea/expression dichotomy. That is, they will be able to incorporate ideas but not protected expression.

The implementation of a statutory exception to reverse engineering is not altogether novel in a Canadian context. A recommendation implicitly in favour of the enactment of such an exception was made in 1984-85 to Parliament prior to the passage of the copyright reforms that were eventually passed in 1988, and which expressly placed computer programs within the jurisdiction of the *Copyright Act*.[350] Although the recommendation suggested that the government further study issues concerning the shared use of sub-programs, the government decided against proceeding with further studies, and no such statutory exemption, however limited, was ever enacted.[351] It should be noted that these recommendations were made in 1984-85 prior to even the basic implementation of computer program protection within the *Copyright Act*. Much has changed, both in terms of technology and the legal protection of new technology, since

---

350 *A Charter of Rights for Creators*, Sub-Committee on the Revision of Copyright, House of Commons, Government of Canada, 1985, at Recommendation 61. The Sub-Committee on the Revision of Copyright recommended that the government should "study the possibility of providing an exception to permit the reproduction of a substantial part of a pre-existing program as a non-substantial part of another program." This would create an exception similar to the reverse engineering exception found in the *ICTA*. The Sub-Committee felt that innovation of computer programs would be accelerated through an exception which allowed the shared use of modular constructs. The Sub-Committee also found that this was a "normal and healthy" practice in the computer industry, and that the law should not impose costs on the industry by preventing the sharing of program code. Although no mention was made of reverse engineering in the copyright revision proposals, it is clear that in order to use sub-programs interchangeably with one's own computer programs, one would need the specifications and parameters to do so. If not provided by the manufacturer of the first computer program, the only way of obtaining these parameters would be through reverse engineering the sub-programs, or modular constructs.

351 Bohdan Romaniuk, "Are Computer Software and Integrated Circuitry Legally Vulnerable to Reverse Engineering - Part One", (1986) 3 Canadian Computer Law Reporter 177, at pp. 178 - 79.

then. The Canadian government's unwillingness to implement the "shared sub-program" exception in the mid-1980s was not accompanied by any statement to the effect that this route would be forever forestalled and its refusal should not be taken as indicating any more than a cautious approach to what were uncharted waters at the time. Furthermore, the "shared sub-program" proposal was far more revolutionary than any reverse engineering exception would ever be since it proposed an exception to the general copyright principle that prohibits the substantial copying of a work. The reverse engineering exception was merely a necessarily incidental effect to the ultimate purpose mandated by that proposal. A reverse engineering exception, as proposed in this paper, would only challenge traditional copyright principles insofar as it would allow intermediate copying to occur; it would not exempt the use of what was uncovered by the process of reverse engineering from traditional copyright principles.

Additionally, the fact that the twelve nations of the European Union will each soon have statutory exceptions in favour of allowing reverse engineering, coupled with the recent American decisions which allow reverse engineering as a fair use to their copyright legislation, no longer makes any such enactment revolutionary. In fact, the reverse may be true. Canada's failure to allow the reverse engineering of computer programs may soon place it in a minority amongst industrialized nations. Clearly the time has come to re-consider the adoption of such a statutory exception.

2. Relying on the Fair Dealing Exception

Another approach to the problem of reverse engineering constituting a technical violation of copyright would be to judicially or legislatively expand upon the, largely untouched, fair dealing defence.[352] Legislative changes to fair dealing have been recommended[353] and subsequently rejected.[354] To legislatively widen fair dealing merely to provide an exception to the reverse engineering of computer programs would presumably encounter similar hostility. Because the purpose of fair dealing is to have a general application to works protected under the *Act*, it would make far better sense to enact a *sui generis* exception, as set out above, than to legislatively widen the fair dealing concept.

A judicial widening of fair dealing, on the other hand, puts the problem entirely in the hands of judges, thus avoiding these difficulties, and provides greater flexibility in applying the exception to reverse engineering than would a legislative exception. Further, a judicial widening of fair dealing does not run the risk of being tied

---

352 The recommendation discussed in this section applies equally to a public interest defence. However, since the public interest defence is not statutory, and has barely received judicial notice in the field of copyright, it has relatively little chance of being applied when compared with fair dealing. See *supra*, note 134, and accompanying text.

353 In *From Gutenberg to Telidon: A White Paper on Copyright*, Consumer and Corporate Affairs, Government of Canada, Ottawa (1984), at Section V), it was recommended that fair dealing, because of its "lack of statutory definition", be replaced with a fair use section containing a "prioritized list of factors to be considered in determining whether a particular use of a work is a fair use". Presumably this list would be similar to that set-out in the U.S. *Act*.

354 *A Charter of Rights for Creators*, Sub-Committee on the Revision of Copyright, House of Commons, Government of Canada, 1985, at Recommendations 82, 83, and 85. The Sub-Committee on the Revision of Copyright recommended that fair dealing be retained and that no prioritized list of factors be enacted as "the flexibility so essential to fair dealing would be destroyed by the fact that they would be mandatory and exhaustive". (At pp. 64 - 65). The Sub-Committee, however, was not against the further use of -illustrative, non-mandatory, factors.

to statutory language which may not presently contemplate future challenges to our potentially limited current thinking.[355] The concept of fair dealing has remained largely untouched by Canadian courts and its breadth remains relatively uncertain. Given our American neighbours' application of their fair use doctrine to reverse engineering cases, it would certainly be no surprise to see Canadian courts expand fair dealing in a similar manner. After all, as copyright finds itself used more and more frequently in situations which were not contemplated by the drafters of the legislation, the fair dealing exception, designed to provide the courts with an instrument of flexibility, should not remain untouched. Fair dealing should be developed and its scope made more certain. In accordance with the language of section 27 of the *Copyright Act*, in order to allow reverse engineering under fair dealing, a court would be forced to qualify the reverse engineering as occurring for the purposes of either "research" or "review", and would conclude that any intermediate copying required in the reverse engineering process is a fair dealing since the ultimate goal of the process is to uncover underlying ideas or unprotectable expression for the purposes of achieving compatibility and standardization.[356]

---

355 For this reason it has been recommended that any statutory exception enacted to address the reverse engineering problem should provide the courts with as much flexibility as possible and be framed in broad terms, avoiding the mention of specific technologies.

356 A recommendation that the "research" objective, as outlined in the fair dealing section (section 27 of the *Copyright Act*), be revised to read "private research" was rejected by the Canadian Parliament (*A Charter of Rights for Creators*, Sub-Committee on the Revision of Copyright, House of Commons, Government of Canada, 1985, at Recommendation 84). The intent of this proposal was to preclude commercial organizations from making use of the fair dealing defence. The rejection of this proposal suggests that commercial organizations are indeed allowed to use the fair dealing defence.

The flip side of the "judicial widening of fair dealing" argument is, of course, that given the reluctance of the courts to develop the fair dealing exception, waiting for a court to apply fair dealing to any new subject matter remains uncertain.[357] Given the juridical history of fair dealing, it is not clear that the courts will attempt to delineate fair dealing any time soon. They have not done so, to any great degree, for seventy years, and there has been no indication given that they are willing to start. Furthermore, relying on courts to use the fair dealing exception to permit reverse engineering will first require that an infringement claim concerning reverse engineering and a defense based on fair dealing be presented to the court. Until such a case is presented the uncertainty associated with reverse engineering and its permissibility under copyright law will create an environment that may potentially dissuade reverse engineering, which we have already determined is a desirable activity. That is, since reverse engineering is *prima facie* a violation of copyright, any artificial impediment, whether it be uncertainty or a flat out prohibition, results in the loss of potential efficiencies and societal wealth. Put another way, should the reverse engineering of computer programs have been expressly permitted, who is to say what products, technologies and standards may have developed. Finally, as mentioned, using fair dealing to permit reverse engineering will potentially not extend to cases where there is a finding of trade secret protection by virtue of section 63 of the *Copyright Act*. If, however, it is determined that trade secret protection, in the traditional sense, constitutes a valid limit

---

357 It is also questionable whether fair dealing can apply where an entire work, as opposed to a part thereof, has been copied. This restriction does not apply to the American fair use exception. See *supra*, note 127, and accompanying text; and *supra* note 273, and accompanying text.

on the scope of allowable reverse engineering then this section will not pose any difficulty. There is a remote possibility, however, that the courts may broadly construe compiled object code as being confidential and thus be unwilling to apply fair dealing as a result of the decision in *Beloff v. Pressdram*.[358] This argument, were it to be successful, would have the effect of granting computer programs some special status (in additional protection) because of their form. While this type of confidentiality protection exists within the realm of trade secrets, it is beyond the intended scope of copyright.

As a result of these limits on fair dealing -- primarily the time it will take to clearly outline the law concerning reverse engineering through the judicial process, and the potentially complex legal issues it will raise in an already complicated area of law -- a clear legislative statement will probably be more successful in effectively resolving the problems associated with reverse engineering computer programs.

3. Alternatives to Copyright

A third, more revolutionary, method of dealing with the problem of reverse engineering is, as many critics have been calling for, to re-draft intellectual property protections as they apply to computer programs. Ever since the protection of computer programs was recognized as a problem, there have been calls for the enactment of *sui generis* legislation designed to deal with the problem. As a clearer understanding of the economic importance of computers has emerged coupled with the law's relative

---

[358] See *supra*, notes 130, 131, and accompanying text.

inability to effectively deal with the subject matter through existing legislation and judicial decision making, the calls for computer program protection, other than copyright, have been received with less skepticism. Unfortunately, the fact that copyright protection of computer programs has gained global acceptance has now largely precluded a complete shift away from copyright. The inertia caused by this effect is not unlike the "going down the wrong path" difficulties with standardization, mentioned previously.

As the differences between computer programs and more traditional copyrightable works become increasingly apparent, and as more legislative exceptions become warranted, it would seem to be a logical step to eventually enact legislation that would supplement the *Copyright Act*, and would cover those aspects that are not appropriate to copyright protection. For example, there have been innumerable difficulties caused in relation to the protection of computer screens. The Canadian government has indicated that it would not afford such protection under its copyright legislation.[359] It has become apparent, however, that under certain circumstances, the courts feel that screens are indeed the proper subject matter of copyright.[360] The legal reasoning presented in the

---

359 In their report to the House of Commons, the Sub-Committee on the Revision of Copyright recommended that "there should be no right of display [with respect to user interface screens] in the revised law". At the time, the government responded by stating that they would study this recommendation in further detail although the first round of amendments to the *Act* did not address these recommendations. *Ibid*, Recommendation 53. This position was further supported by a government report drafted in conjunction with the Uruguay Round of the GATT talks where it was stated that, in the context of trade, copyright ought not to be extended to the "look and feel" or "structure, sequence, and organization" of a program, or to algorithms, ideas, systems, and the like. "Detailed Canadian Proposal on Standards Issues", Department of External Affairs (Multi-lateral Trade Negotiations Office), Government of Canada.

360 See, for example, the decision in *Delrina Corp. v. Triolet Systems Inc.* (1993), 47 C.P.R. (3d) 1 , Court file no. 12515/86 (Ont. Ct. Gen. Div.); and *Gemologists Inc. v. Gem Scan International*

decisions, largely influenced by U.S. case law, is tenuous and does not provide certainty to those in the industry. This is not the fault of the judiciary, who have been given poor tools in the copyright legislation and are, as a result, ill-equipped to couch their decisions in clear and simple terms. A more sound approach would be to supplement the copyright legislation with legislation expressly designed to handle those issues that are unique to computer programs. In the same way, an exception to reverse engineering might more appropriately find its home in such legislation, as reverse engineering is similarly unique to computer programs and does not readily apply itself to more traditional works to which copyright applies. While it is beyond the scope of this paper to present such a *sui generis* regime of computer program protection, one should be nonetheless aware that the possibility of supplementing copyright with other legislation exists.

Related to the possibility of developing entirely new computer program specific legislation is the more realistic possibility that patent protection will become increasingly important as a form of intellectual property protection for computer programs. Protecting computer programs using patent legislation is already a reality and this pattern is continually increasing. Computer programs protected under the patent regime must, of course, make full disclosure of the manner in which they operate. The trade-off for meeting the more rigorous requirements of the patent system is that the monopoly protection granted is much greater. Unlike copyright, even if a patented product is reproduced completely independently of the original product, a royalty must

---

Inc. (1986), 9 C.P.R. (3d) 255 (Ont. H.C.); leave to appeal to the Ont. C.A. refused (1986), 10

still be paid to the patent holder. Although the term for patent protection is considerably less than copyright,[361] for computer programs it is likely that the term will be more than adequate.[362] With patent protection, the need for a reverse engineering right is obviated since full disclosure is mandatory prior to obtaining the patent, and is subsequently made publicly available.

E. The Scope of a Reverse Engineering Right

As mentioned, if an exception to copyright, either statutory or fair dealing in form, is to be applied then it should not be restricted, as was the ruling of the Ninth Circuit in *Atari Games Corp. v. Nintendo of America Inc.*, to the discovery of ideas. Permission of reverse engineering should not be limited to the case where the intended purpose is to uncover ideas. Similarly, reverse engineering should not only be permitted on those parts of a computer program which are necessary to understand the underlying ideas. Instead, permission to reverse engineer a computer program should be construed broadly in favour of the individual seeking to reverse engineer the computer program. Attempting to limit the process to only those parts of a computer program that capture the underlying ideas is conceptually difficult and sometimes impossible. Furthermore, a specific enquiry that attempts to uncover various ideas underlying a program runs the risk of ignoring the big picture of how the ideas interact with one another. This big picture is, in itself, so conceptually abstract that it too properly falls within the realm of ideas.

C.P.R. (3d) 431 (Ont. H.C.).

361 Patents are protected for a period of twenty years from the date of filing the patent under the Canadian *Patent Act* (section 44).

362 Furthermore, the two regimes are not necessarily mutually exclusive in their operation.

Restricting reverse engineering in this manner is wholly consistent with copyright law and its intended purpose of protecting expression and not ideas.

Similarly, reverse engineering seeking to uncover expression should also not be restricted. Seldom will this be the case, however, since idea and expression are so intertwined that those seeking to reverse engineer a computer program will always claim that they are attempting to discover an underlying unprotectable idea, which will no doubt be true to some degree. Judicial intervention to determine otherwise would at best be entirely subjective and would be made by a court that is not expert in matters concerning computers. The uncertainty and probability of erroneous decision-making coupled with the waste of resources in bringing such actions to court is highly questionable. Furthermore, at the end of the day any decision to allow or prohibit reverse engineering deters from the more appropriate inquiry as to whether protectable expression was incorporated into the newly developed computer program. It is extremely difficult for a court to determine *a priori* whether reverse engineering was meant to uncover idea or expression, whereas the probability of making a correct decision when faced with the question as to whether expression has been used in an infringing manner in the development of a new program is relatively much higher. Furthermore there is a convincing argument to be made that any ideas, however small, underlying a computer program are beyond the mandate of copyright protection. If one is to allow reverse engineering to uncover "large" ideas, whatever that may mean, then surely this reasoning must extend to all ideas. The fact that computer programs exist in a naturally encoded

state should not cause the copyright law to prevent the uncovering of ideas whatever their relative importance might be.

If protectable expression is uncovered during reverse engineering and is illicitly used in the construction of a new computer program then this use would violate copyright independently of the act of reverse engineering the computer program.[363] The holder of the first program's copyright would simply bring an action for infringement against the creator of the infringing work. The fact that reverse engineering was performed would not be in issue and this fact would not prejudice the final outcome of the action. Given the potential liability for infringement if copied expression is used in the construction of another computer program, the only obvious explanation for reverse engineering a computer program where expression is sought would be in a situation where the computer program being reverse engineered contains expression which somehow impedes access or compatibility with it, as in the aforementioned lock-out

---

363 This is akin to the first argument made by Accolade in the *Sega Enterprises v. Accolade Inc.* case. In that case, Accolade "maintain[ed] that intermediate copying does not infringe the exclusive rights granted to copyright owners in section 106 of the [U.S.] Copyright Act unless the end product of the copying is substantially similar to the copyrighted work." (*Supra*, note 253, at p. 1565). Accolade lost on this argument based on the fact that the *Sega* Court felt bound by its decision in *Walker v. University Books*, 602 F.2d 859 (9th Cir. 1979) which concerned the intermediate copying of books. The *Walker* decision was framed in broad language which the *Sega* Court felt must be applied to computer programs as well. The Court was, as a result, unwilling to provide a less restrictive interpretation of the fair use doctrine, preferring to allow reverse engineering only where the purpose of the process was to gain an understanding of ideas and purely functional concepts embodied in a computer program which are not protected by copyright. Had the Court decided otherwise, it may have distinguished the *Walker* decision based on the uniquely functional nature of computer programs as compared with more traditional literary works. That is, intermediate copying for books should not properly be equated to intermediate copying for computer programs since the nature of the copying is for altogether different purposes. Oddly enough, this argument is not a far cry from the "purposeful" analysis engaged in by the *Sega* Court in arriving at its decision.

cases.[364] Although it is entirely within the right of a programmer to create such mechanisms and hidden data that allows access, it is not within the intended purpose of copyright law to provide additional legal protection to such devices. If the expression used is found to be protectable, independently of the reverse engineering issue, then an action for infringement will be successful in any event. Attempting to limit the scope of reverse engineering to ideas is both redundant and the risk of poor decision making is too high in relation to the trivial benefits such a rule would confer.

---

364 According to one commentator, "[t]he inadvertant protection of ideas under copyright may suggest that reverse engineering should be permitted in all instances [although it] is particularly justified [in] making a new program compatible with existing copyrighted software. *Supra*, note 35, at pp. 2022 - 23.

Chapter VI. Conclusion

Although Canadian courts have not yet faced a case where the reverse engineering of a computer program has been in issue, it is nonetheless prudent for our legislators to consider the issue prior to a court challenge where the results are bound to be relatively unpredictable. Thus far, at least thirteen other western nations have confronted the problem and have accepted that the reverse engineering of computer programs, under certain circumstances, does constitute a valid exception to copyright infringement. Given the increasing reliance of the Canadian economy on the software industry, it is only a matter of time before the question of reverse engineering gets raised in Canada. Furthermore, by confronting this issue, earlier rather than later, the legislature has an opportunity to accelerate technological progress in the computer software field. This reasoning underlay the formulation of the decompilation provisions in the E.U.'s *Software Directive* and was similarly recognized by the Canadian parliament in its inclusion of the reverse engineering provisions in the *Integrated Circuit Topography Act*.

The case for creating some form of reverse engineering exception arises from the fact that, practically speaking, the only manner in which a computer program may be disassembled constitutes an infringement of the copyright protections granted computer programs under the *Copyright Act*. When examined from an economic perspective, the advantages to allowing reverse engineering far outweigh the disadvantages, and remain consistent with the goals of the copyright legislation, both past and present. Briefly, an economic perspective was chosen as the appropriate lens through

which to examine the issue because in an Anglo-American context, the origins and basis of copyright protection are fundamentally based on economic issues. By applying the copyright legislation as it currently stands, thereby prohibiting reverse engineering, a serious risk of granting a monopoly over functional standards arises. Copyright legislation was designed well before the existence of computer technology, and its application to this field can be characterized as awkward at best. Copyright legislation was not designed to protect ideas or purely utilitarian works. By placing computer programs under the umbrella of copyright, the situation has arisen where the copyright monopoly may now stifle the development of standards and slow the pace of technological advancement, which runs in direct contrast to the stated purpose of copyright law. Preventing the disclosure of such functional processes to the public at large does not exist in any form of intellectual property protection other than copyright as it applies to computer programs.

In the case of the United States, although its legislature has thus far remained silent on the matter, higher U.S. courts have faced reverse engineering issues in several cases and have begun to carve out a reverse engineering exception, for intermediate copying, based on the defense of fair use as found in the U.S. copyright legislation. The reasoning of the 9th Circuit Court of Appeals in the *Sega Enterprises Ltd. v. Accolade Inc.* decision is essentially correct. However, the Court's purposeful analysis, which requires that reverse engineering must be limited to only those elements that are not protected by copyright, falls just short of the mark. Unfortunately, the court felt itself constrained by the language of an earlier decision concerning the intermediate

copying of traditional literary works and did not choose to recognize the special and unique nature of computer programs.

Although the scope of the reverse engineering right under fair use remains relatively restricted as compared with the European Union's legislative provisions, the extent to which American courts will allow reverse engineering to occur has not been fully settled. The Canadian *Copyright Act*'s fair dealing defense, although not as evolved as the fair use defense, presents Canadian jurists with the flexibility to allow the intermediate copying of computer programs necessary for reverse engineering to occur, notwithstanding the fact that such copying is *prima facie* an infringement. Although fair dealing will allow courts to circumnavigate the intermediate copying problem, it would be more suitable for Parliament to implement an express exception to reverse engineering that construes the process in a broad, as opposed to a restrictive, manner. Based on the limited use of the fair dealing exception to-date, relying on the courts to apply fair dealing in such a manner remains an optimistic and uncertain proposition at best.

Construing a statutory exception in a broad manner is wholly consistent with the purpose of copyright protection, and would continue to protect the expression of a work from being copied. This position, however, is revolutionary in light of those exceptions that have thus far been passed. There is a general consensus that reverse engineering be limited to cases where compatibility with a computer program is sought. The difficulty, subjectivity, and futility of such a limitation serves to render the limitation effectively meaningless and, practically speaking, highly uncertain in its application. It is

extremely difficult to determine if one who reverse engineers a computer program truly does so for purposes of compatibility. Furthermore, in cases where reverse engineering occurs, it is logical to assume that no specifications have been provided by the original program designers. In such a case, it is very difficult for the individual performing the reverse engineering to determine *ex ante* exactly what s/he is looking for. To effectively interface with a computer program one must understand its general structure as well as specifics about operational characteristics. A court is in a poor position to decide at what point the reverse engineering process no longer concerns compatibility. Even if such a determination could be made, it would serve little purpose in the context of the reverse engineering process. Instead, the determination of whether compatibility is being achieved should be made when examining the allegedly infringing computer program. At this stage, if it is determined that expression that does not relate to compatibility has been used then a decision of infringement can be rendered. If a program is reverse engineered and is not copied then no loss occurs to the original owners save the exposure of underlying ideas, expression and processes to the reverse engineer. If the ideas and processes are exposed through publication, there is no question that copyright cannot prevent this. If expression is in some way published, then this will infringe the original work. To keep the entire underlying structure of a computer program secret because of its binary form properly falls under the law of trade secrets and is well beyond the mandate of copyright.

Bibliography

A. Legislation & Treaties

Berne Convention, (1886 as amended to 1928) as reprinted in *Copyright Act*, R.S.C. 1985, c.C-42 as amended, Sch. II.

Berne Convention for the Protection of Literary and Artistic Works, September 9, 1886, revised July 24, 1971, 828 U.N.T.S. 221.

Combines Investigation Act, R.S.C., 1927, c. 36.

Competition Act, R.S.C. 1985, c. C-34.

Constitution Act, 1867.

Copyright Act, (1814) (U.K.), 54 Geo. 3, c. 156.

Copyright Act, (1842) (U.K.), 5 & 6 Vict., c. 45.

Copyright Act, (1911) (U.K.), 1 & 2 Geo. 5, c. 46.

Copyright Act, C. 24; R.S.C. 1927, c. 32; R.S.C. 1952, c. 55.

Copyright Act, R.S.C. 1985, c.C-42.

Copyright, Designs and Patents Act 1988, (U.K.), c. 48.

Criminal Code, R.S.C. 1985, c. C-46.

Criminal Code, R.S.C., 1927, c. 36.

European Council Directive of May 14, 1991 on the Legal Protection of Computer Programs (91/250/EEC).

Integrated Circuit Topography Act, R.S.C. 1985, c. I-14.6 [1990, c. 37].

Licensing Act of 1662, (U.K.), 14 Cha. 2, c.33..

Patent Act, R.S.C. 1985, c. P-4.

Rome Convention for the Protection of Performers, Producers of Phonograms and Broadcasting Organizations, October 26, 1961, 496 U.N.T.S. 43.

Semiconductor Chip Protection Act, 17 U.S.C. §§ 901 - 914 (1988).

Statute of Anne, (U.K.), 8 Anne, c. 19.

Treaty Establishing the European Economic Community (*EEC* Treaty) (1958).

Treaty on European Union, February 7, 1992, Europe Documents No. 1759/60 (*The Maastricht Treaty*).

U.S. *Constitution*.

U.S. *Copyright Act*, 17 U.S.C. § 101.

Universal Copyright Convention, adopted on September 6, 1956 in Geneva, as revised in Paris on July 24, 1971.

B. Table of Cases

Aldrich v. One Stop Video Ltd., 13 C.I.P.R. 202, 17 C.P.R. (3d) 27 (B.C.S.C.).

American Restatement of the Law of Torts (1939).

Amoco Canada Petroleum Co. Ltd. v. Texaco Exploration Canada Ltd., [1976] 1 F.C. 258.

An Act to amend the Copyright Act and to amend other Acts in consequence thereof, S.C.

Apple Computer Inc. V. Mackintosh Computers Ltd. (1986), 10 C.P.R. (3d) 1 (F.C.T.D.); varied (1987), 44 D.L.R. (4th) 74 (Fed. C.A.), aff'd [1990] (S.C.C.).

Apple Computer, Inc. v. Microsoft Corporation, 759 F.Supp 1444 (N.D.Cal 1991).

Atari Games Corp. v. Nintendo of America Inc., 975 F.2d 832 (Fed. Cir. 1992).

Autodesk Inc. V. Dyason (1990), 18 I.P.R. 109 (Aust. Fed. Ct.), reversed (1992), A.I.P.C. 90,855, 22 I.P.R. 162 (Aust. H.C.).

Bell Canada v. Intra Canada Communications (1982), 70 C.P.R. (2d) 252 (F.C.A.); rev'g (1982), 62 C.P.R. (2d) 21 (F.C.T.D.).

Beloff v. Pressdram, [1973] 1 All E.R. 241 (Ch. D.).

Betts v. Wilmott (1871), 6 Ch.App. 239 (U.K.C.A.).

Breen v. Hancock House Publishers Ltd. et al. (1986), 6 C.I.P.R. 129 (F.C.T.D.).

Breen v. Hancock House Publishers (1985), 6 C.I.P.R. 129 (F.C.T.D.).

Breeze Corpns. v. Hamilton Clamp & Stampings Ltd., [1962] O.R. 29 (Ont. H.C.).

Broderbund Software Inc. v. Unison World Inc., 648 F.Supp. 1127 (N.D. Cal. 1986).

Canadian Admiral Corp. Ltd. V. Rediffusion Inc, [1954] Ex.C.R. 382.

Cardwell v. Leduc (1962) 23 Fox Pat. C. 99 (Ex.Ct.).

CMAX v. UCR Inc., 4 CCH Computer Cases ¶ 46,752 (U.S. Dist. Ct., Ga., 1992).

Coco v. A.N. Clark Ltd. [1969] RPC 41 (Ch.D).

Compo Co. Ltd. v. Blue Crest Music Inc. (1979) 45 C.P.R. (2d) 1 (S.C.C.).

Computer Associates International, Inc. V. Altai, Inc., 23 USPQ2d 1241 (2nd Cir. 1992).

Creditel of Canada Ltd. v, Faultless (1977) 36 C.P.R. (2d) 88 (Ont. H.C.).

Delrina Corp. v. Triolet Systems Inc. (1993), 47 C.P.R. (3d) 1 , Court file no. 12515/86 (Ont. Ct. Gen. Div.).

Digital Communications Associates, Inc. v. Softklone Distributing Corporation, 659 F.Supp 499 (N.D. Ga. 1987).

Donaldson v. Becket, (1774) 1 E.R. 837; 4 Burr. 2408.

E.F. Johnson Co. v. Uniden Corp. of America, 623 F.Supp. 1485 (D.C. Minn. 1985).

Eli Lilly & Co. v. Marzone Chemicals Ltd. (1976), 29 C.P.R. (2d) 253 (F.C.T.D.); aff'd [1977] 2 F.C. 104.

F & I Retail Systems Ltd. v. Thermo-guard Automotive Products Canada Ltd. et al. (1984), 1 C.P.R. (3d) 297 (Ont. H.C.).

Feist Publications Inc. v. Rural Telephone Service Inc., 499 U.S. 340 (1991).

Francis Day & Hunter Ltd. V. Bron, [1963] Ch. 587 (C.A.).

Geac Canada Ltd. v. Prologic Computer Corp. et al., [1987] Vancouver Registry C872594 (B.C.S.C.).

Gemologists Inc. v. Gem Scan International Inc.(1986), 9 C.P.R. (3d) 255 (Ont. H.C.); leave to appeal to the Ont. C.A. refused (1986), 10 C.P.R. (3d) 431 (Ont. H.C.).

Gondos v. Hardy (1982), 64 C.P.R. (2d) 145 (Ontario. H.C.).

Gottschalk v. Benson, 409, U.S. 63, 175 U.S.P.Q. 672 (1972).

Harper & Row Publishers v. Nation Enterprises, 471 U.S. 539, 105 S.Ct. 2218.

Hubbard v. Vosper, [1972] 1 All E.R. 1023 (C.A.).

Hustler Magazine, Inc. v. Moral Majority, Inc., 796 F.2d 1148 (9th Cir. 1986).

International Corona Resources Ltd. v. LAC Minerals Ltd. (1986) 53 O.R. (2d) 737 (Ont. H.C.).

John Richardson Computers v. Flanders and Chemtec (U.K. High Court, February 19, 1993).

Kewanee Oil Co. v. Bicron Corp., 416 U.S. 470 (1974).

La Societe D'Informatique R.D.G., Inc. v. Dynabec Ltee et al. (1984), 6 C.P.R. (3d) 299 (Que.S.C.).

LAC Minerals Ltd. v. Intern. Corona Res. (1989) 26 CPR (3d) 97 (SCC).

Lamb v. Evans, [1893] 1 Ch. 218 (C.A.).

Lasercomb America Inc. v. Reynolds, 911 F.2d 970 (4th Cir. 1990).

Lewis Galoob Toys Inc. v. Nintendo of America Inc., 20 U.S.P.Q.2d 1662 (D.C. N.D. Ca. 1991).

Lion Laboratories Ltd. V. Evans, [1985] Q.B. 526 (C.A.).

Lotus Development Corporation v. Borland, Inc., 788 F.Supp. 78 (D. Mass. 1992).

Lotus Development Corporation v. Paperback Software International and Stephenson Software Ltd., 740 F.Supp. 37 (D. Mass. 1990).

Massie & Renwick Limited v. Underwriters' Survey Bureau Limited, [1940] S.C.R. 218.

Massie & Renwick Ltd. v. Underwriters' Survey Bureau Ltd., [1940] S.C.R. 218; var'g [1938] Ex. C.R. 103.

Millar v. Taylor, (1769) 98 E.R. 201, 4 Burr. 2303 (K.B.).

Moreau v. St.Vincent, [1950] Ex. C.R. 198.

North American Systemshops Ltd. v. King (1989), 68 Alta. L.R. (2d) 145 (Q.B.D.).

Orbitron Software Design Corp. v. M.I.C.R. Systems Ltd. (1990), 48 B.L.R. 147 (B.C.S.C.).

Philco Products Ltd. v. Thermionics Ltd., [1939] Ex.C.R. 147; aff'd, [1940] S.C.R. 501.

Philco Products Ltd. v. Thermionics Ltd., [1941] Ex. C.R. 209, var'd [1943] S.C.R. 396.

Positron Inc. v. Desroches, [1988] R.J.Q. 1636 (Que. Superior Ct.).

Q-Co. Industries Inc. V. Hoffman, 625 F.Supp. 608 (S.D.N.Y. 1985).

R. V. James Lorimer and Co. Ltd., [1984] 1 F.C. 1065 (F.C.A.).

R.L. Crain Ltd. v. Ashton [1949] 2 D.L.R. 471 (Ont. H.C.), affirmed [1950] O.R. 62 (Ont. C.A.).

RBM Equipment Ltd. v. Philips Electronic Industries Ltd. (1973), 10 C.P.R. (2d) 23 (F.C.).

RBM Equipment Ltd. v. Philips Electronic Industries Ltd., [1973] 1 F.C. 103.

Ridgewood Resources Ltd. v. Hemuset (1982) 18 Alta. L.R. (2d) 68 (Alta. C.A.), leave to appeal to SCC refused (1982) 43 N.R. 90 (S.C.C.).

SAS Institute Inc. v. S & H Computer Systems 605 F.Supp. 1816 (U.S.D.C. 1978).

SAS Institute Inc. v. S & H Computer Systems, 605 F.Supp. 1816 (U.S.D.C., 1985).

Schlumberger Canada Ltd. v. Commissioner of Patents (1981), 56 C.P.R. (2d) 204 (F.C.A.).

Schlumberger Canada Ltd. v. Commissioner of Patents (1981), 56 C.P.R. (2d) 204 (F.C.A.); leave to appeal refused (1982), 62 C.P.R. (2d) 261.

Sega Enterprises Ltd. V. Accolade Inc., 23 U.S.P.Q.2d 1440 (D.C. N.D. Ca. 1992).

Sega Enterprises Ltd. v. Accolade Ltd., 24 U.S.P.Q.2d 1561 (9th Cir. 1992).

Software Solutions Associates v. Depow (1989) 25 C.P.R. (3d) 129.

Sony Corp. v. Universal Studios, 464 U.S. 417; 220 U.S.P.Q. 665 (1984).

Synercom Technology, Inc. v. University Computing Company and Engineering Dynamics, Inc., 462 F.Supp. 1003 (1978).

Systèmes Informatisés Solartronix v. Cégep de Jonquière (1988), 22 C.I.P.R. 101 (Que. Sup. Ct.).

United States v. Loew's, Inc., 371 U.S. 38 (1962).

Universal City Studios v. Sony Corporation of America, 220 U.S.P.Q. 665 (1984).

Table of Cases

University of London Press Ltd. v. University Tutorial Press Ltd. [1916], 2 Ch. 601.

Vault Corp. V. Quaid Software Ltd, 655 F.Supp. 750 (E.D. La. 1987); aff'd, 847 F.2d 255 (5th Cir. 1988).

Walker v. University Books, 602 F.2d 859 (9th Cir. 1979).

Whelan Associates Inc. v, Jaslow Dental Laboratory Inc., 797 F.2d 1222 (3rd Cir. 1986), *cert. denied* 479 U.S. 1031 (1987).

Zamacois v. Douville (1943), 3 Fox Pat. C. 44 (Ex. Ct.).

C. *Secondary Sources*

"Taking the Stand: The Look and Feel Issue Examined", *PC Magazine*, May 26, 1987..

Amory, Hugh. "'*De facto* Copyright'? Fielding's *Works* in Partnership, 1769-1821", (1984) 17 Eighteenth Century Studies 449.

Austin, Richard B.. "Misuse of Copyright", (1991) 8 Canadian Computer Law Reporter 53.

Bainbridge, David I.. "Computer Programs and Copyright: More Exceptions to Infringement", (1993) 56 The Modern Law Review 591.

Becker, Stephen A.. "Drafting Patent Applications On Computer-Implemented Inventions", [Vol. 4, Spring, 1991] Harv.J.L. & Tech 237.

Berg, Jeff. "Moral Rights: A Legal, Historical and Anthropological Reappraisal", (1991) I.P.J. 341.

Brown, Harold R.. "Fear and loathing of the paper trail: originality in products of reverse engineering under the Semiconductor Chip Protection Act as analogized to the fair use of nonfiction literary works", (1990) 41 Syracuse Law Review 985.

Burkill, Guy. "Reverse compilation of computer programs and its permissibility under the Berne Convention", (1990) 6 Computer Law & Practice 114.

Byrne, J.D.. "Computer programs and reverse engineering: Recent European Developments", (1991) 8 Canadian Computer Law Reporter 45.

Calabresi, G.. "Some Thoughts on Risk Distribution and the Law of Torts", (1961) 70 Yale L.J. 499.

Canada, Government of. "Detailed Canadian Proposal on Standards Issues", Department of External Affairs (Multi-lateral Trade Negotiations Office), Government of Canada.

Canada, Government of. *A Charter of Rights for Creators*, Sub-Committee on the Revision of Copyright, House of Commons, Government of Canada, 1985.

Clifford Chance, "Computer and Communications Bulletin", March 1993, Issue 9.

Clifford Chance, "The European Software Directive" (Clifford Chance, U.K. 1991).

Cline, Dennis. "Copyright protection of software in the EEC: the competing policies underlying community and national law and the case for harmonization", (1987) 75 California Law Review 633.

Colombe, Michael, and Meyer, Caroline. "Seeking interoperability: an industry response", (1990) 12 E.I.P.R. 79.

Consumer & Corporate Affairs Canada - Department of Communications, "Semiconductor Chip Protection in Canada: Proposals for Legislation", Government of Canada, Department of Supply & Services, April, 1987.

Cornish, W.R.. *Intellectual Property: Patents, Copyright, Trade Marks and Allied Rights* (London: Sweet & Maxwell, 1989).

Earle, Edward. "The Effect of Romanticism on the 19th Century Development of Copyright Law", (1991) 6 I.P.J. 269 (1991).

Firestone, Robert F.. "Copyright Infringement of Non-literal Elements of Computer Programs: 'Substantial Similarity', But Not Uniformity, In The Law", (1992) 1*Mealey's Litigation Reports Intellectual Property* 49.

Fisk, George and Clark, Jane. "Hardware and Software Protection in Canada", (1990) 10 Computer Law Journal 483.

Fox, H.G.. *The Canadian Law of Copyright and Industrial Designs*, 2nd edition (Toronto: Carswell, 1967).

From Guttenberg to Telidon: A White Paper on Copyright, Consumer and Corporate Affairs, Government of Canada, Ottawa (1984).

Fugere, Kathryn A.. "Reverse engineering under the Semiconductor Chip Protection Act: an argument in favor of a 'value-added' approach" (Symposium: Intellectual Property Law), (1992) 22 Golden Gate University Law Review 515.

Handa, Sunny and Buchan, James. "Copyright as it Applies to the Protection of Computer Programs in Canada", (1994) I.I.C. [pending publication].

Handa, Sunny. "Have Recent Copyright Decisions Unduly Suppressed the Emergence of Standardized User Interfaces" (Paper written at the Faculty of Law, University of Toronto, 1991) [unpublished].

Hau, Zhou. "Securing Patent Protection for Computer Program-Related Inventions", (March, 1993) Patent World 34.

Secondary Sources

Hayes, David L.. "The Legality of Disassembly of Computer Programs", (1993) 12 Computer/Law Journal 1.

Hebert, Monique. *Copyright Act Reform*, Research Branch, Library of Parliament, Government of Canada, 1982 reviewed 1987.

Hogg, Peter W.. *Constitutional Law of Canada*, (Toronto: Carswell, 1985).

Ignatin, Gary R.. "Let the Hackers Hack: Allowing the Reverse Engineering of Copyrighted Computer Programs to Achieve Compatibility", 140 University of Pennsylvania Law Review 1999.

Johnson-Laird, Andrew, "Reverse engineering of software: separating legal mythology from actual technology" (Intellectual Property Symposium), (1992) 5 Software Law Journal 331.

Karjala, Dennis S.. "Copyright, Computer Software, and the New Protectionism", (1987) 28 Jurimetrics J. 33.

Kasch, Stephen P.. "The Semiconductor Chip Protection Act: Past, Present and Future", (1992) 7 High Technology L.J. 72.

Knable Gotts, Ilene. "Regulators Focusing on Antitrust Issues", The National Law Journal (January 24, 1994), p. S12.

Landes, William and Posner, Richard A.. "An Economic Analysis of Copyright Law", (1989) 189 Journal of Legal Studies 325.

Lipsey, R. *et al.*. *Economics 5th ed.*, (Harper & Row: New York, 1982).

Mackaay, Ejan. "Economic Incentives in Markets for Information and Innovation", (1990) 13 Harvard Journal of Law & Public Policy 867.

Mackaay, Ejan. "Informational Goods: property of a mirage", (1985) 1 Computer Law and Practice 193.

Mann, J. Fraser. *Computer Technology and the Law in Canada* (Toronto: Carswell, 1987).

McGregor, Sheila. "Look & Feel - Australia", (1993) 1 Mealey's Litigation Reports 44.

Meiners, Roger E. & Staff, Robert J.. "Patents, Copyrights, and Trademarks: Property or Monopoly", (1990) 13 Harvard Journal of Law & Public Policy 911.

Menell, Peter S.. "Scope of Copyright Protection for Programs", 41 Stan.L.Rev. 1045.

Secondary Sources

Michelman, Frank. "A Comment on Some Uses and Abuses of Economics in Law", (1979) 46 The University of Chicago L.R. 307.

Miller, Arthur R.. "Copyright Protection for Computer Programs, Databases, and Computer-Generated Works: Is Anything New Since CONTU?", (1993) 106 Harvard Law Review **?**.

Miller, Clifford G.. "The Proposal for an EC Council Directive on the Legal Protection of Computer Programs" (1990) 12 Eur. Intell. Prop. Rev. 347.

Moore, Harold C.. "Atari v. Nintendo: Super Mario Uses "Expressive" Security Feature to "Lock" Out the Competition", (1992) 18 Rutgers Computer & Technology L.J. 919.

Morrison, Linda. "The EC Directive on the Legal Protection of Computer Programs: Does it Leave Room for Reverse Engineering Beyond the Need for Interoperability?", (1992) 25 Vanderbilt J. of Transnational Law 293.

David Nimmer *et al.*. "A Structured Approach to Analyzing the Substantial Similarity of Computer Software in Copyright Infringement Cases", (1988) 20 Ariz. St. L.J. 625.

Palmer, Tom G.. "Are Patents and Copyrights Morally Justified? The Philosophy of Property Rights and Ideal Objects", (1990) 13 Harvard Journal of Law & Public Policy 817.

Panko, Raymond R.. *End User Computing*, (New York: John Wiley & Sons, 1988).

Patry, William F.. *The Fair Use Privilege in Copyright Law* (U.S.A.: The Bureau of National Affairs Inc., 1985).

Posner, Richard A.. "Some Uses and Abuses of Economics in Law", (1979) 46 The University of Chicago L.R. 281.

Posner, Richard A.. "The Economic Approach to Law", (1975) 53 Texas L.R. 757.

Reed, Chris. "Reverse engineering computer programs without infringing copyright", (1991) 13 E.I.P.R. 47.

Rhodes, Michael. "Prohibiting the decompilation of software" (Editorial), (1993) 9 Computer Law & Practice (United Kingdom) 1.

Rice, David A.. "Public Goods, Private Contract and Public Policy: Federal Preemption of Software Licence Prohibitions Against Reverse Engineering", (1987) 53 Univ. of Pitt. L.R. 543.

Secondary Sources

Robertson, Ranald. *Legal Protection of Computer Software* (London: Longman Law, 1990).

Romaniuk, Bohdan. "Are Computer Software and Integrated Ciruitry Legally Vulnerable to Reverse Engineering - Part One", (1986) 3 Canadian Computer Law Reporter 177.

Rose, Mark. "The Author as Proprietor: *Donaldson v. Becket* and the Genealogy of Modern Authorship", (1988) 23 Representations 51.

Ruston, David A.. "Bus drivers and other conductors - a review of semi-conductor topography protection", (1990) 6 Canadian Intellectual Property Review 155.

Schmidtz, David. "Contracts and Public Goods", (1987) 10 Harvard Journal of Law & Public Policy 475.

Schnell, Angelika, and Freska, Anna M.. "On the threshold of 1992 Europe debates software protection; report on the International Symposium on Reverse Engineering in West Berlin March 1989" (1990) 6 Santa Clara Computer and High-Technology Law Journal 59.

Sinnott, Timothy J.. "Copyright in operating system software on computer chips: a tale of two apples", (1987) 3 Intellectual Property Journal 1.

Sookman, Barry B.. *Computer Law: Acquiring and Protecting Information Technology* (Toronto: Carswell, 1989).

Stein, Michael D.. "The Importance of a Trade Secret as a Supplement to Copyright Protection of Computer Software", (Fall 1993) 12 I.P.L. Newsletter 28.

Stephen, Frank H.. *The Economics of the Law* (Ames: Iowa State University Press, 1988).

Stern, Richard H.. "An ill-conceived analysis of reverse engineering of software as copyright infringement: Sega Enterprises Ltd. v. Accolade, Inc.", (1992) 14 E.I.P.R. 407.

Stern, Richard H.. "Reverse engineering of software as copyright infringement - an update: Sega Enterprises Ltd. v. Accolade, Inc.", (1993) 15 E.I.P.R. 34.

Stewart, S.M.. *International Copyright and Neighbouring Rights*, 2nd. ed. (London: Butterworths, 1989).

Trudeau, Raymond. "Software Patents", (1992) 9 C.I.P.R. 234.

Secondary Sources

United States Congress, Office of Technology Assessment, *Intellectual Property Rights in an Age of Electronics and Information* (Report), OTA-CIT-302 (Washington, DC: U.S. Government Printing Office, April, 1986).

U.S. House and Senate Explanatory Memoranda, 130 Cong. Rec. 28,960.

U.S. National Commission of New Technological Uses of Copyrighted Works (CONTU), *Final Report* (1978).

Victor, David. "An Analysis of an Affirmative Defense for Reverse Engineering Within a System of Legal Protection for Computer Software", (1993) 66 Southern California L.R. 1705.

Von Lewinski, Silke. "Copyright in the European Communities: The Proposed Harmonization Measures, (1992) 18 Brooklyn Journal of International Law 703.

Weichselbaum, Mindy J.. "The EEC directive on the legal protection of computer programs and U.S. copyright law: should copyright law permit reverse engineering of computer programs", (1991) 14 Fordham International Law Journal 1027.
7