



**Stanford – Vienna
Transatlantic Technology Law Forum**

*A joint initiative of
Stanford Law School and the University of Vienna School of Law*



TTLF Working Papers

No. 1

**A Status Report from the Software
Decompilation Battle:**

**A Source of Sores for Software Copyright
Owners in the United States and the
European Union?**

Petra Heindl

2008

TTLF Working Papers

About the TTLF Working Papers

TTLF's Working Paper Series presents original research on technology, and business-related law and policy issues of the European Union and the US. The objective of TTLF's Working Paper Series is to share "work in progress". The authors of the papers are solely responsible for the content of their contributions. The TTLF Working Papers can be found at <http://tflf.stanford.edu>. Please also visit this website to learn more about TTLF's mission and activities.

If you should have any questions regarding the TTLF's Working Paper Series, please contact Vienna Law Professor Siegfried Fina, Stanford Law Professor Mark Lemley or Stanford LST Executive Director Roland Vogl at the

Transatlantic Technology Law Forum
<http://tflf.stanford.edu>

Stanford Law School
Crown Quadrangle
559 Nathan Abbott Way
Stanford, CA 94305-8610

University of Vienna School of Law
Department of Business Law
Schottenbastei 10-16
1010 Vienna, Austria

Sponsors

This project was co-sponsored by the Stanford-Vienna Transatlantic Technology Law Forum (Stanford Law School/University of Vienna School of Law), the Stanford Center for E-Commerce, and the Forum on Contemporary Europe at the Freeman Spogli Institute for International Studies at Stanford University.

About the Author

Petra Heindl received her Mag.iur. [JD] from the University of Vienna School of Law in Austria. After graduating from the Vienna Law School, she completed an LL.M. in European Union law at the Danube University Krems in Austria. In addition, she earned an LL.M. in U.S. Law from Santa Clara University School of Law.

Petra Heindl is a Fellow of the Stanford-Vienna Transatlantic Technology Law Forum as well as a PhD candidate of the Vienna Technology Law Program at the University of Vienna School of Law, Austria. Her research work is also connected with the Forum on Contemporary Europe at Stanford University's Freeman Spogli Institute for International Studies. She works as an associate with Wolf Theiss Attorneys at Law in Vienna, Austria. Her research focuses on European and international business and technology law.

General Note about the Content

The opinions expressed in this paper are those of the author and not necessarily those of the Transatlantic Technology Law Forum or any of its partner institutions, or the sponsors of this research project

Suggested Citation

This TTLF Working Paper should be cited as:

Petra Heindl, A Status Report from the Software Decompilation Battle: A Source of Sores for Software Copyright Owners in the United States and the European Union?, TTLF Working Paper No. 1, http://www.law.stanford.edu/program/centers/ttlf/papers/heindl_wp1.pdf.

Copyright

© 2008 Petra Heindl

Abstract

For traditional media, such as novels, copyright represents a “bargain” between the individual author and the general public: the author has an exclusive right to make and sell copies, but anyone can look at the novel, learn from its ideas, and use those ideas as a stimulus for the creation and a reward for the publication of new works. However, computer code by its unique nature, to wit, its invisibility, its not readily accessibility, makes it a quite unconventional subject matter for enjoying copyright protection all around the world. Thus, for software, which is generally available only as object code, the “bargain” is one-sided: the author gets a monopoly, but the public does not have access to even the uncopyrightable ideas contained in the code. Unless an explicit exception recognizes otherwise, copying of source code would violate the copyright holder's exclusive rights, even if the person who decompiled the code only used it as an interim step in gaining access to the program's functionality or developing a new, independent computer program. Given the vast profitability of computer software reverse engineering in general, and decompilation in particular, has evolved to a battle over the right of competitors to reverse engineer, including the necessity of intermediate copies of an original computer software. In the wake of establishing an identity in the global computing world, especially to be able to form a counterpart to the U.S. dominance in the market, the European Communities adopted the 1991 Directive on the Legal Protection of Computer Programs. With its decompilation exception the Directive manifests Europe's position in the overall reverse engineering battle, as one towards “open systems.” Yet the U.S. Copyright Act of 1976 does not contain any express provision on legitimizing intermediate copying in the course of software decompilation. Consequently, the U.S. legislatures left it up to the judiciary to define the policy on the subject. More recent case law, as well as the Digital Millennium Copyright Act indicates a shift in U.S. copyright law toward a “closed” proprietary model of software development. In the face of EC case law, such as *Magill*, *IMS Health* and *Microsoft*, there is a perception that the European Commission considers the protection afforded by national copyright laws to go beyond the legitimate reward for the software author's creative and financial efforts. It is in cases involving intellectual property rights where there is the greatest danger of misuse of an essential facility. While U.S. software providers may invoke the copyright misuse defense, European case law – devoid of a copyright misuse doctrine or alike – refers to the essential facility doctrine for software providers to apply in cases where national copyright laws provide for near-monopoly protection up to an anti-competitive extent. Under European law, it is competition law that may oblige a dominant owner of the essential facility, including computer code, to cooperate with its competitors on competition grounds. In so doing, courts have relied on basic principles of antitrust economics of the impact of refusal to deal. A similar balancing test, which is yet neither undisputed nor uniformly applied by U.S. judges, has been established under U.S. case law when referring to the doctrine of copyright misuse. To date, the misuse defense has not been codified in the Copyright Act; neither has the U.S. Supreme Court yet clarified the existence and scope of copyright misuse. This reluctance indicates another concession to enhanced protection of software copyright owners in opposition to reverse engineering, and thereby continues to pursue a “closed source code” model. By responding to the refusal of access to uncopyrightable

interface information contained in copyrighted computer software with a compulsory licensing order the European Court of First Instance in the *Microsoft* case has recently given further direction to an “open software” model in Europe. With the background of copyright policy considerations and the current legal treatment of software decompilation in both the EC and the United States, this research paper will not only re-consider Europe’s position recently expressed in the *Microsoft* case but also point out possible ramifications of the compulsory licensing order imposed on Microsoft regarding its Window’s interface information for the legal status of decompiling computer software in Europe as well as in the United States.

To My Mom and Dad

REFERENCES

I. Books

- Abrams, Howard B., The Law of Copyright, Egan, MN: Thomson/West, 2002.
- Albors-Llorens, Albertina. EC Competition Law and Policy, Cullompton, Portland: Willan Publishing, 2002.
- Anderman, Steven D., & Kallaugher, John. Technology Transfer and the New EU Competition Rules: Intellectual Property Licensing after Modernisation, Oxford: University Press, 2006.
- Bainbridge, David I. Software Copyright Law, London: Pitman Publishing (1992).
- Bandey, Brian. International Copyright in Computer Program Technology, Birmingham: CLT Professional Publishing, 1996.
- Bishop, Simon, & Walker, Mike. The Economics of EC Competition Law: Concepts, Application and Measurement, London: Sweet & Maxwell Ltd., 2002 (2nd ed.).
- Broder, Douglas F. A Guide to US Antitrust Law, London: Sweet & Maxwell, 2005.
- Chisum, Donald S., Chisum on Patents: A Treatise on the Law of Patentability, Validity and Infringement, New York, NY: M. Bender, 1978 –.
- Cotter, Thomas F. “Intellectual Property and the Essential Facilities Doctrine” in The Economics of Intellectual Property, Towse, Ruth, & Holzhauser, Rudi (eds.), Celtenham, UK; Northampton, USA: Edward Elgar Publishing, 2002 (Volume IV).
- Czarnota, Bridget, & Hart, Robert J. Legal Protection of Computer Programs in Europe—A Guide to the EC Directive, London, Dublin, Edinburgh, Munich: Butterworths, 1991.
- Drexl, Josef. What is protected in a Computer Program?: Copyright Protection in the United States and Europe, Weinheim, New York: VCH, 1994.
- Gellhorn, Ernest, & Kovacic, William E., & Calkins, Stephen, Antitrust Law and Economics in a Nutshell, St. Paul, MN: Thomson/West, 2004 (5th ed.).
- Goldstein, Paul. Goldstein on Copyright, New York, NY: Aspen Publishers, 2007 (3rd ed.).

- Haberstumpf, Helmut. “Der urheberrechtliche Schutz von Computerprogrammen“ in Rechtsschutz und Verwertung von Computerprogrammen, Lehmann, Michael (ed.), Köln: Verlag Dr. Otto Schmidt KG, 1993 (2nd ed.), at 69 et seq.
- Hildebrand, Doris. The Role of Economic Analysis in the EC Competition Rules, The Hague, London, New York: Kluwer Law International, 2002 (2nd ed.).
- Jaburek, Walter J. Das neue Software-Urheberrecht, Wien: Verlag Medien und Recht, 1993.
- Jongen, Herald D.J., & Meijboom, Alfred P. (eds.). Copyright Software Protection in the EC, Deventer-Boston: Kluwer Law and Taxation Publishers, 1993.
- Lehmann, Michael (ed.). Rechtsschutz und Verwertung von Computerprogrammen, Köln: Verlag Dr. Otto Schmidt KG, 1993 (2nd ed.).
- Marly, Jochen. Urheberrechtsschutz für Computersoftware in der Europäischen Union, München: C.H. Beck'sche Verlagsbuchhandlung, 1995.
- Moritz, Hans-Werner, & Tybusseck, Barbara. Computersoftware: Rechtsschutz und Vertragsgestaltung—Eine fächerübergreifende Darstellung nach deutschem und EG-Recht, München: Verlag C.H. Beck, 1992 (2nd ed.).
- Nimmer, Melville B., & Nimmer, David. Nimmer on Copyright, New York, NY: M. Bender, 2007.
- Rodger, Barry J., & MacCulloch, Angus. Competition Law and Policy in the European Community and United Kingdom, London, Sydney: Cavendish Publishing Limited, 1999.
- Schuhmacher, Wolfgang. Die Essential Facility Doctrine als Herausforderung für das Urheberrecht, in Dittrich, Robert. Beiträge zum Urheberrecht VII, at 87 et seq., Wien: Manz'sche Verlags- und Universitätsbuchhandlung, 2003.
- Thomann, Felix H., & Rauber, Georg (eds.). Softwareschutz, Bern: Stämpfli, 1998.
- Ullrich, Hanns, & Körner, Eberhard. Der internationale Softwarevertrag, Heidelberg: Verlag Recht und Wirtschaft GmbH, 1995.
- Van den Berg, Roger J., & Camesasca, Peter D. European Competition Law and Economics: A Comparative Perspective, Antwerpen, Groningen, Oxford, New York: Intersentia, 2001.
- Verstrynge, J. F. “Protecting Intellectual Property Rights Within the New Pan-European Framework—Computer Software”, Paper presented at the World Computer Law Congress 1991 in Los Angeles 10 (Apr. 18-28, 1991), *reprinted in* Rechtsschutz und Verwertung von Computerprogrammen, Lehmann, Michael (ed.), Köln: Verlag Dr. Otto Schmidt KG, 1993 (2nd ed.).

Walter, Michel M. (ed.). Europäisches Urheberrecht, Wien: Springer Verlag, 2001.

II. Periodicals

Andrews, Jeffrey A. “Reversing Copyright Misuse: Enforcing Contractual Prohibitions on Software Reverse Engineering” 41 Hous. L. Rev. 975 (2004).

Bayha, Betsy E. “Reverse Engineering of Computer Software in the United States, the European Union, and Japan” C137 ALI-ABA 175 (1995).

Blythe, Stephen E. “The U.S. Digital Millennium Copyright Act and the E.U. Copyright Directive: Comparative Impact on Fair Use Rights” 8 Tul. J. Tech. & Intell. Prop. 111 (2006).

Brooks, Ernie L., & Cunningham, Thomas. “Reverse Engineering in U.S., Japanese, and European Economic Community Copyright Law” SD72 ALI-ABA 191 (1999).

Carver, Brian W. “Share and Share Alike: Understanding and Enforcing Open Source and Free Software Licenses” 20 Berkeley Tech. L.J. 443 (2005).

Donahoe, Robert V. “Does Intermediate Copying of Computer Software for the Purpose of Reverse Engineering a Non-Infringing Product Infringe the Copyright in the Software?” 2001 B.C. Intell. Prop. & Tech. F. 111301 (2001).

Dorenkamp, Theodore. “Copyright Misuse or A Right To Compete?: A Critique of Alcatel USA v. DGI Technologies” 9 Tex. Intell. Prop. L. J. 269 (2001).

Dreier, Thomas. “The Council Directive of May 14 1991 on the Legal Protection of Computer Programs” 9 Intell. Prop. Rev. 319 (1991).

Ehrlich, Marc A. “Fair Use or Foul Play? The EC Directive on the Legal Protection of Computer Programs and Its Impact on Reverse Engineering” 13 Pace L. Rev. 1003 (1994).

Fallenbock, Markus. “On the Technical Protection of Copyright: The Digital Millennium Copyright Act, the European Community Copyright Directive and Their Anticircumvention Provisions” 7 Int’l J. Comm. L. & Pol’y 4 (2002).

Fellmeth, Aaron Xavier. “Copyright Misuse and the Limitations of the Intellectual Property Monopoly” 6 J. Intell. Prop. L. 1 (1998).

Frischmann, Bret, & Moylan, Dan. “The Evolving Common Law Doctrine of Copyright Misuse: A Unified Theory and Its Application to Software” 15 Berkeley Tech. L. J. 865 (2000).

- Gage, Joe L. Jr. "Copyright Law and Reverse Engineering: Have Recent Decisions Taken the Fair Out of Use?" 46 Baylor L. Rev. 183 (1994).
- Gardiner, Merritt A. "Bowers v. Baystate Technologies: Using the Shrinkwrap License to Circumvent the Copyright Act and Escape Federal Preemption" 11 U. Miami Bus. L. Rev. 105 (2003).
- Gaster, Jens-L. "Der Erfahrungsbericht zur EG-Computerprogramm-Richtlinie" Computer & Recht 1/2000, 11 (2000).
- Gilbert-Macmillan, Kathleen. "Intellectual Property Law for Reverse Engineering Computer Programs in the European Community" 9 Santa Clara Computer & High Tech L. J. 247 (1993).
- Guillou, Celine M. "The Reverse Engineering of Computer Software in Europe and the United States: A Comparative Approach" 22 Colum.-VLA J.L. & Arts 533 (1998).
- Haaf, Joseph. "The EC Directive on the Legal Protection of Computer Programs: Decompilation and Security for Confidential Programming Techniques" 30 Colum. J. Transnat'l L. 401 (1992).
- Hartzog, Neal. "Gaining Momentum: A Review of Recent Developments Surrounding the Expansion of the Copyright Misuse Doctrine and Analysis of the Doctrine in its Current Form" 10 Mich. Telecomm. Tech. L. Rev. 373 (2004).
- Heidinger, Roman. "The Essential Facilities Doctrine and Intellectual Property: Compulsory Licenses under EC Competition Law" MR-Int 2006, 23 (2006).
- Hoeren, Thomas. "The EC Directive on Software Protection—A German Comment" 1991 Computer Law & Practice 246 (1991).
- Homiller, Daniel P. "Patent Misuse in Patent Pool Licensing: From National Harrow to 'The Nine No-Nos' to Not Likely" 2006 Duke L. & Tech. Rev. 7 (2006).
- Howard, John V., & Gushow, Jil. "Interoperability in the United States and Europe" 24 Colo. Law. 1275 (1995).
- Johnson-Laird, Andrew. "Software Reverse Engineering in the Real World" 19 U. Dayton L. Rev. 843 (1994).
- Judge, Kathryn. "Rethinking Copyright Misuse" 57 Stan. L. Rev. 901 (2004).
- Karjala, Dennis S. "Copyright Protection of Computer Software, Reverse Engineering, and Professor Miller" 19 U. Dayton L. Rev. 975 (1994).

- Knight, Jennifer R. "Copyright Misuse v. Freedom of Contract: And the Winner Is..." 73 Tenn. L. Rev. 237 (2006).
- Lee, Donna L. "Reverse Engineering of Computer Programs under the DMCA: Recognizing a 'Fair Access' Defense" 10 Marq. Intell. Prop. L. Rev. 537 (2006).
- Lewis, Terril. "Reverse Engineering of Software: An Assessment of the Legality of Intermediate Copying" 20 Loy. L.A. Ent. L. Rev. 561 (2000).
- Linhoff, Joe. "Video Games and Reverse Engineering: Before and After the Digital Millennium Copyright Act" 3 J. Telecomm. & High Tech. L. 209 (2004).
- Loren, Lydia Pallas. "Slaying the Leather-Winged Demons in the Night: Reforming Copyright Owner Contracting with Clickwrap Misuse" 30 Ohio N.U. L. Rev. 495 (2004).
- Mahajan, Anthony J. "Intellectual Property, Contracts, and Reverse Engineering After ProCD: A Proposed Compromise for Computer Software" 67 Fordham L. Rev. 3297 (1999).
- McClanahan, John Baker. "Copyright Misuse as a Defense in an Infringement Action: Lasercomb America, Inc. v. Reynolds" 49 Wash. & Lee L. Rev. 213 (1992).
- Meninsky, Carla. "Locked Out: New Hazards of Reverse Engineering" 21 J. Marshall J. Computer & Info. L. 591 (2003).
- Morrison, Linda G. "The EC Directive on the Legal Protection of Computer Programs: Does It Leave Room for Reverse Engineering Beyond the Need for Interoperability?" 25 Vand. J. Transnat'l L. 293 (1992).
- Note. "Clarifying the Copyright Misuse Defense: The Role of Antitrust Standards and First Amendment Values" 104 Harv. L. Rev. 1289(1991).
- O'Rourke, Maureen A. "Drawing the Boundary Between Copyright and Contract: Copyright Preemption of Software License Terms" 45 Duke L. J. 479 (1995).
- Prestin, Derek. "Where To Draw the Line Between Reverse Engineering and Infringement: Sony Computer Entertainment, Inc. v. Connectix Corp." 3 Minn. Intell. Prop. Rev. 137 (2002).
- Pruitt, David N. "Beyond Fair Use: The Right to Contract Around Copyright Protection of Reverse Engineering in the Software Industry" 6 Chi-Kent J. Intell. Prop. 66 (2006).
- Reichman, J. H. "Goldstein on Copyright Law: A Realist's Approach to a Technological Age" 43 Stan. L. Rev. 943 (1991).

- Rotenberg, Boris. “The Legal Regulation of Software Interoperability in the EU” NY Jean Monnet Working Paper No. 07/2005 (2005).
- Rothman, Ivan. “From Sega to Sony and Beyond: An Alternative Legal Basis for Software Reverse Engineering” 18 NO. 3 Intell. Prop. L. Newsl. 1 (2000).
- Samuelson, Pamela, & Scotchmer, Suszanne. “The Law and Economics of Reverse-Engineering” 111 Yale L.J. 1575 (2002).
- Soma, John T., & Black-Keefer, Sharon K., & Smith, Alexander R. “Minimizing Reverse Engineering: Sample Language for Dual United States and European Union Software Licenses” 24 Denv. J. Int’l L. & Pol’y 145 (1995).
- Son, Seungwoo. “Can Black Dot (Shrinkwrap) Licenses Override Federal Reverse Engineering Rights?: The Relationship Between Copyright, Contract, and Antitrust Laws” 6 Tul. J. Tech. & Intell. Prop. 63 (2004).
- Spoor, Jaap H. “Copyright Protection and Reverse Engineering of Software: Implementation and Effects of the EC Directive” 19 U. Dayton L. Rev. 1063 (1994).
- Sullivan, Jeffrey D., & Morrow, Thomas M. “Practicing Reverse Engineering in an Era of Growing Constraints under the Digital Millennium Copyright Act and Other Provisions” 14 Alb. L. J. Sci. & Tech. 1 (2003).
- Vinje, Thomas C. “Die EG-Richtlinie zum Schutz von Computerprogrammen und die Frage der Interoperabilität” 4 GRUR Int. 250 (1992).
- Webb, Jere M., & Locke, Lawrence A. “Intellectual Property Misuse: Developments in the Misuse Doctrine” 4 Harv. J. L. & Tech. 257 (1991).
- 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?” 14 Fordham Int’l L. J. 1027 (1991).
- Williams, John A. “Can Reverse Engineering of Software Ever Be Fair Use? Application of Campbell’s “Transformative Use” Concept” 71 Wash. L. Rev. 255 (1996).

III. EU Legislative Materials

Green Paper on Copyright and the Challenge to Technology – Copyright Issues Requiring Immediate Action, COM(88) 172 final, Brussels, June 7, 1988.

Opinion of the Economic and Social Committee on the Green Paper on Copyright and the Challenge to Technology – Copyright Issues Requiring Immediate Action, O.J. (C 71) 9, Brussels, March 20, 1989.

Proposal for a Council Directive on the Legal Protection of Computer Programs, COM(88) 816 final.
— Commission Preparatory Memorandum, O.J. (C 91) 4, Brussels, Apr. 12, 1989.

Opinion of the Economic and Social Committee on the Proposal for a Council Directive on the Legal Protection of Computer Programs, O.J. (C 329) 4, Brussels, Dec. 30, 1989.

Amended Proposal for a Council Directive on the Legal Protection of Computer Programs, COM(90) 509 final.
— Commission Preparatory Memorandum, O.J. (C 320) 22, Brussels, Dec. 20, 1990.

Council Common Position with a view to the adoption of a Directive on the Legal Protection of Computer Programs, (EEC) No. 10652/1/90, Brussels, Dec. 14, 1990.

Council Directive of 14 May 1991 on the Legal Protection of Computer Programs, 1991 O.J. (L 122) 42, Brussels, May 17, 1991.
(as amended by Council Directive 93/98/EEC)

Report from the Commission to the Council, the European Parliament and the Economic and Social Committee on the Implementation and Effects of Directive 91/250/EEC on the Legal Protection of Computer Programs, COM(2000) 199 final, Brussels, Apr. 10, 2000.

CASES

— U.S. case law

Acorn Structures v. Swantz, 846 F.2d 923 (4th Cir. 1988).

Alcatel U.S.A., Inc. v. DGI Technologies, Inc., 166 F.3d 772 (5th Cir. 1999), *reh 'g and reh 'g en banc denied*, 180 F.3d 267 (5th Cir. 1999).

Alfred Bell & Co. v. Catalada Fine Arts, Inc., 191 F.2d 99 (2nd Cir. 1951).

Altera Corp. v. Clear Logic, Inc., 424 F.3d 1079 (9th Cir. 2005).

A&M Records, Inc. v. Napster, Inc., 239 F.3d 1004 (9th Cir. 2001).

Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984).

Assessment Technologies of Wisconsin L.L.C. v. WIREdata, Inc., 350 F.3d 640 (7th Cir. 2003).

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

Aymes v. Bonelli, 47 F.3d 23 (2d Cir. 1995). 980 F.2d 857 (2d Cir. 1992).

Baker v. Selden, 101 U.S. 99 (1880).

Bateman v. Mnemonics, Inc., 79 F.3d 1532 (11th Cir. 1996).

Bleistein v. Donaldson Lithographing Co., 188 U.S. 239 (1903).

Bowers v. Baystate Technologies, Inc., 320 F. 3d 1317 (Fed. Cir. 2003), *cert. denied*, 539 U.S. 928 (2003).

Broadcast Music, Inc. v. Columbia Broadcasting System, 441 U.S. 1 (1979).

Broadcast Music, Inc. v. Hearst/ABC Viacom Entertainment Services, 746 F. Supp. 320 (S.D.N.Y. 1990).

Campbell v. Acuff-Rose Music, Inc., 510 U.S. 569 (1994).

Comb v. Paypal Inc., 218 F. Supp.2d 1165 (N.D. Cal. 2002).

Computer Associates International, Inc. v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992).

Data General Corp. v. Grumman System Support Corp., 36 F.3d 1147 (1st Cir. 1994).

Davidson Associates, Inc. v. Internet Gateway, 334 F. Supp.2d 1164 (E.D. Mo. 2004), *aff'd sub nom.* Davidson & Associates v. Jung, 422 F.3d 630 (8th Cir. 2005).

DSC Communications Corp. v. DGI Technologies, Inc., 898 F. Supp. 1183 (N.D. Tex. 1995);
— 81 F.3d 597 (5th Cir. 1996)
— Alcatel USA, Inc. v. DGI Technologies, Inc., 166 F.3d 772, *reh'g and reh'g en banc denied*, 180 F.3d 267 (5th Cir. 1999).

Edelman v. N2H2, Inc., 263 F. Supp.2d 137 (D. Mass. 2003).

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

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

Forrest v. Verizon Communications, Inc., 805 A.2d 1007 (D.C. Ct. App. 2002).

Harper & Row, Publishers, Inc. v. Nation Enterprises, 471 U.S. 539 (1985).

Hughes v. McMenammon, 204 F. Supp.2d 178 (D. Mass. 2002).

Independent Ink, Inc. v. Trident, Inc., 210 F. Supp.2d 1155 (C.D. Cal. 2002), *aff'd in part, rev'd in part, sub nom.* Independent Ink, Inc. v. Illinois Tool Works, Inc., 396 F.3d 1342 (Fed. Cir. 2005), *cert. granted*, 126 S. Ct. 1291.

In re Napster, Inc. Copyright Litigation, 191 F. Supp.2d 1087 (N.D. Cal. 2002).

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

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

Lexmark International Inc. v. Static Control Components, Inc., 387 F.3d 522 (6th Cir. 2004).

MAI Systems Corp. v. Peak Computer Inc., 991 F.2d 511 (9th Cir. 1993).

Mallinckrodt, Inc. v. Mediaport, Inc., 976 F.2d 700 (Fed. Cir. 1992).

Moore v. Microsoft Corp., 741 N.Y.S.2d 91 (N.Y. App. Div. 2002).

Morrissey v. Proctor & Gamble Co., 379 F.2d 675 (1st Cir. 1967).

Morton Salt Co. v. G. S. Suppiger Co., 314 U.S. 488 (1942).

Motion Picture Patents Co. v. Universal Film Manufacturing Co., 243 U.S. 502 (1917).

M. Witmark & Sons v. Jensen, 80 F. Supp. 843 (D. Minn. 1948).

National Car Rental Systems, Inc. v. Computer Associates International, Inc., 991 F.2d 426 (8th Cir. 1993).

NEC Corp. v. Intel Corp., 645 F. Supp. 590 (N.D. Cal. 1986).

Nichols v. Universal Pictures Corp., 45 F.2d 119 (2d Cir. 1930).

Practice Management Information Corp. v. American Medical Association, 121 F.3d 516 (9th Cir. 1997), *cert. denied*, 118 U.S. 339 (1997), *opinion amended*, 133 F.3d 1140 (9th Cir. 1998).

ProCD, Inc. v. Zeidenberg, 86 F.3d 1447 (7th Cir. 1996).

Salinger v. Random House, Inc., 650 F. Supp. 413 (S.D.N.Y. 1986), *rev'd*, 811 F.2d 90 (2d Cir. 1987), *cert. denied*, 484 U.S. 890 (1987).

SAS Inst., Inc. v. S & H Computer Systems, Inc., 605 F. Supp. 816 (M.D. Tenn. 1985).

Saturday Evening Post Co. v. Rumbleseat Press, 816 F.2d 1191 (7th Cir. 1987).

Secure Services Tech. v. Time and Space Processing, 722 F. Supp. 1354 (E.D. Va. 1989).

Sega Enterprises, Ltd. v. Accolade Corp., 977 F.2d 1510 (9th Cir. 1993).

Sony Computer Entertainment, Inc. v. Connectix Corp., 203 F.3d 596 (9th Cir. 2000), *cert. denied*, 531 U.S. 871 (2000).

Sony Corporation of America v. Universal City Studios, Inc., 464 U.S. 417 (1984).

State ex rel. Summa v. Starke Circuit Court, 238 Ind. 204 (149 N.E.2d 541) (Ind. 1958).

Taquino v. Teledyne Monarch Rubber, 893 F.2d 1488 (5th Cir. 1990).

United States v. Elcom Ltd., 203 F. Supp. 2d 1111 (N.D. Cal. 2002).

United States v. Loew's, Inc., 371 U.S. 38 (1962).

United States v. Microsoft, 65 F. Supp. 2d 1 (D.D.C. 1999).

United States v. Paramount Pictures, Inc., 334 U.S. 131 (1948).

Universal City Studios, Inc. v. Corel, 273 F.3d 429 (2d Cir. 2001).

Universal City Studios, Inc. v. Reimerdes, 111 F. Supp. 2d 294 (S.D.N.Y. 2000).

USM Corp. v. SPS Technologies, Inc., 694 F.2d 505 (7th Cir. 1982).

Vault Corp. v. Quaid Software Ltd., 847 F.2d 255 (5th Cir. 1988).

Video Pipeline, Inc. v. Buena Vista Home Entertainment, Inc., 342 F.3d 191 (3d Cir. 2003), *cert. denied*, 540 U.S. 1178 (2004).

Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222 (3d Cir. 1986).

Wrench LLC v. Taco Bell Corp., 256 F.3d 446 (6th Cir. 2001).

— **EC case law**

British Leyland Motor Corp. & Others. V. Armstrong Patents Company Ltd. & Others (1986) 2 W.L.R. 400 (Feb. 27, 1986) (U.K.)

Buchhaltungsprogramm, Entscheidungen des Bundesgerichtshofes [BGH CR] 1993, 752 (F.R.G.).

Holzhandelsprogramm, Entscheidungen des Bundesgerichtshofes [BGH CR] 1994, 275 (F.R.G.).

Inkassoprogramm, Bundesgerichtshof [BGH] [Federal Court of Justice] May 9, 1985 (I ZR 52/83), Entscheidungen des Bundesgerichtshofes in Zivilsachen [BGHZ] [Federal Supreme Court] 94 (F.R.G.), translated in 17 IIC 1986, 681.

Decision of the Commission of the European Communities of 24 March 2004, Case COMP/C-3/37.792.

Judgment of the European Court of First Instance (Grand Chamber) of 17 September 2007, Case T-201/04 [2005] ECR (not yet published).

— Case T-201/04 [2005] ECR II-1491 (intervention)

— Case T-201/04 R [2004] ECR II-4463 (interim relief – Article 82 EC Treaty)

— Case T-201/04 R [2004] ECR II-2977 (interim relief – intervention).

Table of Contents

I.	Introduction	1
II.	The Infringing Character of Decompilation—A Global Analysis	7
1.	<i>The Unique Nature of Computer Programs</i>	7
1.1.	The Challenge	7
1.2.	What is Computer Code?	7
1.3.	Copyright Protection of Interfaces	9
1.3.1.	The Idea/Expression Dichotomy	9
1.3.2.	The Originality Standard	10
1.3.2.1.	Different Levels of Originality in Europe	10
1.3.2.2.	Originality under U.S. Copyright Law	12
1.3.2.3.	The Originality Standard of the EU Software Directive	13
2.	<i>Decompilation and Intermediate Copying</i>	15
3.	<i>Copyright Policy Considerations of Decompiling Computer Software.....</i>	20
3.1.	A Challenge for Traditional Copyright Policy	20
3.1.1.	Innovation and Consumer Welfare	20
3.1.2.	Exclusive Copyrights	22
3.2.	Purposes of Decompiling Computer Programs	23
3.3.	Is Article 6 Fair to Common Law Traditions in Europe?	24
III.	Software Decompilation under EU Copyright Law	26
1.	<i>The European Version of the Reverse Engineering Battle</i>	26
1.1.	The United States' Involvement: A Fight for "The Last Bastion of American Technological Superiority"?	28
1.2.	Europe's Shot for Open Systems	29
2.	<i>The White Flag: Article 6 of the EU Software Directive</i>	30
3.	<i>A Compromise Called "Interoperability"</i>	32
3.1.	Interoperability—A General Definition	33
3.2.	How to Develop Software Interoperability—An Example	34
3.3.	Interoperability from an Economic Point of View	34
3.3.1.	Horizontal Interoperability	35
3.3.2.	Vertical Interoperability	36
3.4.	The Scope of Interoperability in a European Understanding	37
3.5.	Is there Guidance by the Berne Convention, TRIPS or the WIPO Panel?	40
3.5.1.	The Three-Step Test under Article 9(1) of the Berne Convention	40
3.5.2.	The Three-Step Test Revisited by the WIPO Panel	41
3.5.3.	The Three-Step Test under Article 6(3) of the EU Software Directive	43
3.6. ...	Decompilation beyond the Need for Interoperability	46
3.6.1.	Maintenance of Computer Programs	46

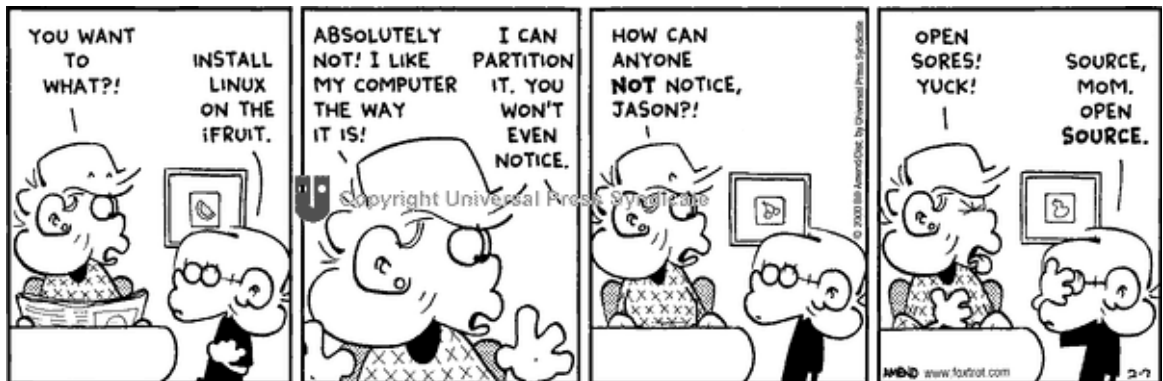
3.6.2.	Error Correction	47
3.6.3.	Research	50
3.7.	Copyright Protection of Know How as Unintended Consequence of Interoperability	51
4.	<i>The Indispensability Criterion</i>	52
4.1.	Analyzing Preparatory Design Material	54
4.2.	Black-Box Analysis (Article 5(3) of the EU Software Directive) ...	56
4.3.	The Burden of Proving Indispensability	59
5.	<i>Legitimate Users</i>	61
6.	<i>Previously Readily Availability</i>	62
7.	<i>Necessary Parts of the Original Program</i>	66
8.	<i>The Restricted Use of Decompiled Information</i>	69
8.1.	Substantial Similarity of Expression	69
8.2.	The “Clean Room” Procedure—A Safeguard?	73
8.3.	The Merger Doctrine under European Copyright Law	75
8.4.	Re-Distribution of Information	78
IV.	Reverse Engineering under U.S. Copyright Law: Let the Battle on the Video Game Market Begin!	79
1.	<i>The “Limited Use” Exception</i>	79
1.1.	Section 117(a) of the Copyright Act	79
1.2.	Vault Corp. v. Quaid Software Ltd.	80
1.3.	The Decision	81
1.4.	A Critical Analysis	82
1.5.	The Response of the Ninth Circuit	84
1.6.	Conclusion	85
2.	<i>The Fair Use Defense</i>	86
2.1.	Section 107 of the Copyright Act	87
2.2.	Atari Games Corp. v. Nintendo of America Inc.	89
2.2.1.	The Facts	89
2.2.2.	The Decision	90
2.2.3.	Fair Use Analysis	90
2.2.4.	The “Legitimate Owner” Requirement	91
2.3.	Sega Enterprises, Ltd. v. Accolade, Inc.	92
2.3.1.	The Facts	92
2.3.2.	The Decision	93
2.3.3.	Fair Use Analysis	93
2.3.4.	Conclusion	95
2.3.5.	Unpublished Source Code	96
2.4.	Sony Computer Entertainment, Inc. v. Connectix Corp.	97
2.4.1.	The Facts	97
2.4.2.	The Decision	98
2.4.3.	How “Transformative” is the Sony Ruling Compared to the Sega Concept?	99
2.5.	The Indispensability Criterion	103
2.5.1.	Indispensability under the Sega Rule	103
2.5.2.	Indispensability under the Sony Rule	104

2.5.3.	Indispensability under the EU Software Directive	106
2.6.	The Role of Interoperability in U.S. Case Law	107
2.6.1.	A Contemporary Analysis of Software Interoperability	107
2.6.2.	Shall Future Compatibility be Protectible?	108
2.7.	Two Different Standards of “Necessity”—One Unnecessary Confusion	109
2.8.	Decompilation of Computer Hardware	111
2.8.1.	U.S. Case Law	111
2.8.2.	The European Approach	112
V.	The Reverse Engineering Exception under the Digital Millennium Copyright Act	114
1.	<i>A Piece of Legislation to Keep Pace with the Technological Advancement</i>	<i>114</i>
1.1.	WIPO’s Influence on the Battle between Hollywood and Silicon Valley	114
1.2.	The Anti-Circumvention Provisions under Section 1201(a) of the DMCA	115
1.3.	The Reverse Engineering Exception	116
1.3.1.	Section 1201(f)(1) of the DMCA	116
1.3.2.	The Scope—A Bill of Conditions	117
1.4.	Sony v. DMCA	118
2.	<i>A Fair Re-Balance under the DMCA?</i>	<i>120</i>
2.1.	The Pre-DMCA Balance	120
2.2.	The Post-DMCA Balance	120
2.2.1.	Interoperability	121
2.2.2.	Focusing on Circumventing Acts	122
2.2.3.	Legalized Copyright Misuse under the Auspices of the DMCA?	122
3.	<i>The EU Software Directive meets Section 1201(f)(1) of the DMCA: Same Wording, but Different Policies</i>	<i>125</i>
VI.	Another Strategy in the Decompilation Battle: Shrink-wrap and Click-on Agreements	126
1.	<i>Contractual Restrictions under the EU Software Directive</i>	<i>127</i>
1.1.	Article 9(1) of the EU Software Directive	127
1.2.	Protection under Trade Secret Law	130
2.	<i>Shrink-wrap and Click-on Agreements under U.S. Copyright Law</i>	<i>131</i>
2.1.	Pre-emption by Federal Copyright Law	132
2.1.1.	Express Pre-emption under Section 301(a) of the Copyright Act	132
2.1.2.	Conflict Pre-emption under the Supremacy Clause	134
2.2.	Restrictive Licensing Practices as Copyright Misuse	135
2.3.	The Doctrine of Unconscionability	140

VII. A Transatlantic Comparison in Re Software Decompilation	142
1. <i>What Happened So Far: Fair Use, Copyright Misuse and Article 6 of the EU Software Directive</i>	<i>142</i>
1.1. Is the Fair Use Concept Really Fair?	142
1.2. Copyright Misuse as a Fairer Legal Basis for Software Decompilation under U.S. Law?	145
1.2.1. The Doctrine of Intellectual Property Misuse	145
1.2.2. Pre-Lasercomb: The Patent Misuse Doctrine	147
1.2.2.1. Morton Salt Co. v. G.S. Suppiger Co.	147
1.2.2.2. Post-Morton Salt Developments	148
1.2.2.3. Legislative Reflections in Patent Law	149
1.2.3. Lasercomb and its Progeny: The Copyright Misuse Defense	150
1.2.3.1. Lasercomb America, Inc. v. Reynolds (1990) ...	150
1.2.3.2. The Practice Management Decision (1997)	151
1.2.3.3. The Alcatel Decision (1999)	152
1.2.3.4. The Napster Decision (2002)	153
1.2.2.5. The Video Pipeline Decision (2003)	155
1.2.2.6. The WIREdata Decision (2003)	156
1.2.4. Two Standards—One Copyright Misuse Doctrine ...	157
1.2.4.1. The Public Policy Rationale	157
1.2.4.2. The Antitrust-Like Standard	159
1.2.5. Procedural Nature of the Misuse Defense	163
1.2.6. Current Impact of the Intellectual Property Misuse Doctrines	163
1.2.6.1. Patent Misuse Doctrine	164
1.2.6.2. Copyright Misuse Doctrine	165
1.3. Contract Law and Decompilation	168
2. <i>The Future: Compulsory Licensing of Interface Information as Sui Generis Copyright Remedy?</i>	<i>171</i>
2.1. Unconventional Consequences of Unconventional Circumstances	171
2.2. Interoperability at the Crossroads of Copyright and Competition Laws	173
2.3. The General Concept of the Essential Facility Doctrine	178
2.3.1. U.S. Antitrust Law	179
2.3.2. EC Competition Law	181
2.4. The Essential Facility Doctrine Applied to the Incentive/Access Tradeoff	182
2.5.1. Relevant Case Law	184
2.5.1.1. Magill	184
2.5.1.2. IMS Health	185
2.5.2. The Bifurcated Role of Disclosure under Copyright Law	187
2.5.3. Microsoft	189
2.5.3.1. The Facts	189
2.5.3.2. The Judgment	191
2.5.4. Highlights of the Microsoft Judgment	193

2.5.4.1.	Shall Future Compatibility Be Protectible?	193
2.5.4.2.	The “Ultima Ratio” Approach Revisited: Compulsory Licensing	194
3.	<i>Compulsory Licensing of Intellectual Property Rights</i>	197
3.1.	Compulsory Licensing as the Fairest Solution?	197
3.2.	Microsoft’s Perspective	201
VIII.	Conclusion	204

I. Introduction—May I Introduce: Jason And His Mom!



© Universal Press Syndicate

What Jason’s Mom probably does not know is that the nature of “open source” software lies in disclosure of the source code.¹ What Jason probably does not know is that the source code embodies copyrightable as well as non-protectible components of a computer program. Disclosing—and thereby enabling anyone to copy—source code thus may turn out to be a software copyright proprietor’s “open sores”.

As a consequence, commercial computer software²—open source software in particular will be set aside for the purpose of this paper—is usually distributed only in its unintelligible object code, so-called “closed software”.³ Nonetheless, software programmers can derive the way the source code works by taking the object code in their possession and, in working backwards, either reverse engineer the program

¹ Generally there are three ways of how to disclose source code. Some software developers publish their source codes (i.e. the so-called “open source” software), some license them freely, and others maintain their source codes as closely held trade secrets.

² Software is a generic term to describe a collection of one or more computer programs. “Software” encompasses a broader scope than “computer program”. Software includes not only the computer program itself but also materials associated with the development and operation of a computer program, the so-called program documentation, but not computer hardware. See HABERSTUMPF in LEHMANN, *Rechtsschutz und Verwertung von Computerprogrammen* [Legal Protection and Use of Computer Programs] (1993) 76-7.

³ Open Source Software refers to software that is distributed with its source code, thus allowing users to easily use, copy, and distribute software, either verbatim or with modifications, either gratis or for a fee, as users deem fit. By contrast, closed software is distributed without the source code, which is thus kept secret. If source code can lawfully be decompiled and disassembled, hence the computer programs are potentially “open”.

For a discussion of the history and implications of open source software, see, e.g., Brian W. Carver, *Share and Share Alike: Understanding and Enforcing Open Source and Free Software Licenses*, 20 BERKELEY TECH. L. J. 443 (2005).

by using the techniques of decompilation (or disassembly)⁴ or conduct an analytical “clean room” operation. In either process, a program’s object code is first copied and then transformed into a human-readable computer code, the source code. In addition to intermediate copying of the original computer program (phase 1: decompilation)⁵, the process of reverse engineering⁶ also involves building the source code program based on the decompiled information of the original (phase 2: implementation).⁷ Decompiling object code produces an approximation of the original source code, known as “intermediate copy”.

Why have courts and legislators around the world been struggling with the legality of acts of decompilation, and subsequent reverse engineering, under copyright law?⁸ Part II will give the answer by uncovering the legal interface of copyright protection and decompilation of computer programs. Throughout the world, computer programs are protected by copyright law. Copyright law provides a bundle of exclusive rights to the software programmer; among others, the right to reproduce and to translate the original work. Without permission of the original software copyright owner acts of decompilation constitutes a breach of these rights granted to the software programmer. Copyright protects the expression of an idea

⁴ In legal terms, the technical differences of decompilation and disassembly do not matter because both techniques are caught by the meaning of copying. *See* DAVID I. BAINBRIDGE, *SOFTWARE COPYRIGHT LAW* (1992) 109. Thus, for the purpose of this paper decompilation and disassembly *per definitionem* (*see infra* notes 67, 68) are used as synonyms being subject to the same legal treatment.

⁵ Pamela Samuelson & Suzanne Scotchmer, *The Law and Economics of Reverse Engineering*, 111 *Yale L.J.* 1575, 1586-7 (2002).

⁶ The expression “reverse engineering” originates from trade secret law, which allows secrets to be uncovered and applied in own products, if this is achieved by means of analyzing the product which is subject to trade secret protection. Not only copyright law when applied to computer programs but also many patent laws allow a sort of moderate reverse engineering as the exclusive rights of the patent holder do not include acts exclusively aimed at researching the patented product or process. Furthermore, Article 15 of the EC Semiconductor Chip Protection Directive permits reverse engineering of the lay-out of another’s chip product. *See* HARALD D.J. JONGEN & ALFRED P. MEIJBOOM (EDS.), *COPYRIGHT SOFTWARE PROTECTION IN THE EC* (1993) 15.

⁷ Software reverse engineering comprises of making intermediary copies and, to a much larger degree, of an additive process. The reverse engineer starts with the lowest possible level of abstraction devoid of any higher level information, and then adds personal knowledge and experience. Besides its cost factor the result of the reverse engineering process depends heavily upon a reverse engineer’s skill and experience. *See* Andrew Johnson-Laird, *Software Reverse Engineering in the Real World*, 19 *U. Dayton L. Rev.* 843, 901 (1994).

⁸ For an overview of the legal status of reverse engineering under U.S., EU and Japanese copyright law, *see, e.g.*, Betsy E. Bayha, *Reverse Engineering of Computer Software in the United States, the European Union, and Japan*, C137 ALI-ABA 175 (1995); Ernie L. Brooks, Thomas Cunningham, *Reverse Engineering in U.S., Japanese, and European Economic Community Copyright Law*, SD72 ALI-ABA 191 (1999).

in the form of a program, hence protecting the software programmer's investment in writing the software. However, ideas and principles underlying the programmer's software are not copyrightable. For traditional media, such as novels, copyright represents a "bargain" between the individual author and the general public: the author has an exclusive right to make and sell copies, but anyone can look at the novel, learn from its ideas, and use those ideas as a stimulus for the creation and a reward for the publication of new works. However, computer code by its unique nature that is its invisibility and thus its not readily accessibility, makes it a fairly "unconventional" subject matter for enjoying copyright protection. For software, which is generally available only as object code, therefore, the "bargain" is one-sided: the author gets a monopoly but the public has no access to even the uncopyrightable ideas and principles contained in the computer code. Unless an explicit exception recognizes otherwise, copying of source code would violate the programmer's exclusive rights. This will be true, even if the person who decompiled the code only used it as an interim step in gaining access to the program's functionality or developing a new, independent computer program. As reverse engineering, including decompilation, has become a common practice particularly in the software industry, the battle over the right of competitors to decompile computer software is neither unique to the European Communities nor to the United States. For the time being, software decompilation is held to be legal under certain conditions in both the European Communities and the United States. Following the introduction to the technical and legal background of the decompilation dilemma under copyright law, Part III shows how interoperability between all "Jason's Moms" and "Jasons" of the world is reached in the copyright laws of the European Community. Featured by Europe's traditionally strong policy in favor of open and interoperable systems, as early as in 1991, the European Community led the world by adopting a carefully devised decompilation provision⁹

⁹ The effect of the Directive had farther-reaching effects—not only as regards the Community but also in a wider context in Europe as a whole. So the EFTA countries (European Free Trade Association)—Finland, Island, Norway, Austria, Sweden and the Switzerland—as incorporated into a European Economic Area (EEA) adopted the EU Software Directive in their national legislations as binding *acquis communautaire*. To a less formal extent, also then Non-Member States, like Poland, Hungary and the former Czechoslovakia, signed association agreements guaranteeing to meet the Directive's standard regarding the protection of intellectual property rights, thereby to bring about a situation where intellectual property protection in these countries would be

in Article 6 of the Directive on the legal protection of computer programs for adopting by its Member States.¹⁰ The Directive introduced for the first time in any legislation an express right to decompile parts of a computer program to obtain information necessary for the purposes of interoperability.

Part IV of this paper outlines the legal status of decompilation under U.S. copyright law. Contrary to the European approach, the U.S. Copyright Act of 1976 does not contain any direct comment on legitimizing the reverse engineering of software in general, or the intermediate copying in particular.¹¹ Instead the U.S. legislators have left it with the judiciary to define the policy on the subject. The decompilation issue has come up in several cases involving interoperability of video game components. Despite the legal fact that intermediate copying of software is *per se* an infringement of the software copyright owner's exclusive reproduction right pursuant to section 106(1) of the U.S. Copyright Act of 1976, case law as of 1992 and the years following shows a propensity that an analysis of the infringement liability of the intermediate copyist should be colored by considerations of copyright policy and fairness. The Ninth Circuit affirmatively applied the fair use defense to redress the above-mentioned imbalance, holding that reverse engineering is fair use when the defendant only extracted unprotected elements, such as ideas, from the copyrighted computer program. The judicial and legislative solutions relied heavily on the policy behind copyright law in new and previously unaddressed circumstances. The U.S. conception of fair use is by its nature dynamic rather than static a doctrine which makes it perceptible to embrace new circumstances with respect to software technology. As for fair use, the United States presently stands alone in the world intellectual property community. Even European countries of the common law tradition, for example the United Kingdom, rely heavily these days on legislative action—as in the case of the United

progressively aligned in substance with that of the Community. The Directive, moreover, served as model piece of legislation for many other countries' legislative acts worldwide. See Jens-L. Gaster, *Der Erfahrungsbericht zur EG-Computerprogramm-Richtlinie [The After-Action Report on the EC Software Directive]*, COMPUTER & RECHT [COMPUTER & LAW] 1/2000, 11, 15 (2000) (F.R.G.).

¹⁰ Council Directive of 14 May 1991 on the Legal Protection of Computer Programs, 1991 O.J. (L 122) 42 et seq. (as amended by Council Directive 93/98/EEC) [hereinafter EU Software Directive].

¹¹ The U.S. Copyright Act does not include a specific exception for interoperability equivalent to Article 6 of the EU Software Directive. Nonetheless, in other technological contexts, such as the development of mask works for semiconductor chips, Congress has expressly approved the practice of reverse engineering. See 17 U.S.C. § 906(a).

Kingdom, mostly on EU Directives—to keep the copyright law abreast of current developments. Thus far the legality of decompiling computer software in the United States and the EU appears to be very similar in substance.

In the United States, the Copyright Act of 1976 was not created with digital technology in mind. In 1998, the U.S. Congress drafted the Digital Millennium Copyright Act (DMCA) to deal with copyright in the digital world. With the DMCA the United States have become one of the worst countries as far as the freedoms of computer users are concerned. Reminiscent to Article 6 of the EU Software Directive, the reverse engineering exception laid down in section 1201(f) of the DMCA allows circumvention of technological measures for the purpose of reverse engineering to achieve interoperability of an independently created computer program. The question remains whether the DMCA will be capable of properly reconciling the rights of software copyright owners and the privileges of users, the role traditionally played by the fair use doctrine. Part V concludes that—opposed to the mainstream in Europe favoring open systems—the DMCA appears to strengthen the protection of U.S. copyright owners in the reverse engineering community.

License agreements may also bind software users to operate a program in a certain way and to avoid using decompilation techniques on that program. So Part VI turns its focus from public legislative acts to private legislative acts, so-called shrink-wrap and click-on license agreements when used—or misused?—by software providers individually to contract away the software user's decompilation rights. Unlike Article 9(1) of the EU Software Directive that contains of a bright-line rule prohibiting generously contractual restrictions on decompilation, U.S. courts have addressed the enforceability of these clauses generally found in shrink-wrap and click-on agreements to be enforceable. Under the shelter of contract law, U.S. software copyright owners have been found support in adding to these agreements clauses that seek to oust express statutory limitations on the exclusive rights granted to them in the Copyright Act and the DMCA. In so holding, broad enforcement of these shrink-wrap and click-on agreements does not only lead to stagnation in the creation of new, innovative computer programs but also furthers

“closed software”, and thereby taking an opposed position to Europe’s “open system” model.

Europe stands for open systems in order to open its market to the international software industry. Microsoft software illustrates the need for the decompilation exception from the European standpoint. Microsoft software has become an industry standard and thus facing very little competition. Without permitting reverse engineering, including decompilation, of the Microsoft software to be legal, smaller companies, many of which are European, would not have the budget to create a completely innovative competing program, since they essentially would have to reinvent the wheel in order to survive vis-à-vis Microsoft dominating the computer market. That is the reason why the European legislators, basically known for its “frantic” protection of authors, have revolutionarily narrowed the rights of authors of computer programs to such an extent that it permits a software providers to discover and use interface information compiled by the original programmer for the sole purpose of creating a competing substitute. With the background of copyright policy considerations and the current legal treatment of software decompilation in both the EU and the United States, Part VII of this paper reconsiders Europe’s position recently expressed in the *Microsoft* judgment by which a compulsory licensing order regarding Window’s interface specifications was imposed on Microsoft. With the big picture of the current legal status of and the policy dilemma inherent in the decompilation of computer programs in mind, this paper finally concludes that the remedy of compulsory licensing of computer code due to its “unconventional” nature should be considered as the “fairest” solution in decompilation battle.

II. The Infringing Character of Decompilation—A Global Analysis

1. The Unique Nature of Computer Programs

1.1. *The Challenge*

For traditional media, such as novels, copyright represents a “bargain” between the individual author and the general public: the author has an exclusive right to make and sell copies, but anyone can generally identify the unprotectible components of a work through mere observation. For example, the reading of a copyrighted novel enables the reader to discover its underlying ideas and principles, learn from its ideas, and use those ideas as a stimulus for the creation and a reward for the publication of new works.¹² But in the case of a copyrighted computer program, the object code form, as a series of “ones” and “zeros”, in general bars access to those uncopyrightable ideas and principles underlying a program which may not already be determined by studying the mere performance of a program.¹³ In other words, computer code by its unique nature that is its invisibility and thus its not readily accessibility makes it an “unconventional” subject matter for enjoying copyright protection all around the world. For computer software, which is generally available only as object code, the “bargain” is one-sided: the author gets a monopoly by way of denying the general public access to both the copyrighted elements and the uncopyrightable ideas contained in the computer code.

1.2. *What Is Computer Code?*

¹² See II GOLDSTEIN ON COPYRIGHT, GOLDSTEIN ON COPYRIGHT § 7.2.1.4.c, at 7:50-1 (2006) (illustrating the practice of “managed copying” applicable to contexts other than software decompilation: in order to extract the work’s unprotectible elements—for example, the plot of a novel—and, subsequently, to incorporate these unprotected elements in a rival novelist’s work, the lawfully obtained copy of a copyrighted novel simply must be read without making a copy of its protected expression, thereby leading to further works of creativity); see also NIMMER ON COPYRIGHT, *supra* note 12, § 13.05[D][4], at 13-230.17 (2006); see also *Sega*, 977 F.2d at 1520 (referring to the “unique nature of computer object code” as reason for considering reverse engineering cases on a case-by-case basis under section 107 of the 1976 Copyright Act).

¹³ II GOLDSTEIN ON COPYRIGHT, *supra* note 12, § 7.2.1.4.c, at 7:50 (stating that the non-copyrightable ideas and principles underlying copyrighted computer code are far less accessible than those embedded in traditional copyright works).

The source code as expressed in specialized alphanumeric language is the form in which computer software is written and thus is a form incomprehensible to the computer to execute the program.¹⁴ Object code is written in a digital machine language consisting only of “zeros” and “ones”¹⁵ and which thus is intelligible only to a computer while typically incomprehensible to humankind.¹⁶ For a computer to be capable to execute the program the source code must be converted into object code form by using a computer program called “compiler” or “assembler”. Compiling or assembling a computer program means converting a high-level language source code program into object code, being the machine code which can be directly understood by the computer.¹⁷ The reverse engineer then may imprint a permanent version of the reverse engineered program onto a silicon chip for commercial distribution which can be used and operated without the original source code.

Software providers pay particular attention to lock away the source code in the “copyright proprietor’s vault”.¹⁸ Can software authors protect their computer code to an extent as to prevent competitive software developers from scrutinizing the program’s non-protectible ideas and principles embodied in computer programs?¹⁹ The legal battles fought in either jurisdiction illustrate the complex nature of separating idea and expression in works of utility that contain expressive elements. Courts will allow protection for the artistic features of an interface while denying protection to functional feature.²⁰

¹⁴ II GOLDSTEIN ON COPYRIGHT, *supra* note 12, § 7.2.1.4.c, at 7:50.

¹⁵ Object code is expressed by binary numbers. A compressed form of binary is called hexadecimal. Thus, binary can be transformed to unstructured sequence of hexadecimal notation. *See* HABERSTUMPF in LEHMANN, *supra* note 2, at 77; *see also* 4 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 13.05[D][4], at 13-230.17 (2006).

¹⁶ *But see* HANS-WERNER MORITZ & BARBARA TYBUSSECK, COMPUTERSOFTWARE: RECHTSSCHUTZ UND VERTRAGSGESTALTUNG [Computer Software: Legal Protection and Contract Formation] (1992) 66 (maintaining that object code may well be read and understood by humankind even without previously transforming it back into source code; thereby limiting decompilation to a mere facility rather than an indispensable means to access to interface information).

¹⁷ BAINBRIDGE, *supra* note 4, at 109.

¹⁸ 4 NIMMER ON COPYRIGHT, *supra* note 15, § 13.05[D][4], at 13-230.18 (2006).

¹⁹ 17. U.S.C. § 102(b) (1990); EU Software Directive, *supra* note 10, art. 1(2), at 44.

²⁰ *See, e.g., Apple Computer, Inc. v. Microsoft Corp.*, 799 F. Supp. 1006 (N.C. Cal. 1992).

Major aspects of a computer program which are of interest to other programmers of software (or hardware) are the interfaces of the target computer program.²¹ The interface of the computer program to be decompiled includes the form of structure of the files it creates and uses, its allocation and management of computer memory, its interaction with peripheral devices, and its input and output formats.²²

1.3. Copyright Protection of Interfaces

1.3.1. The Idea/Expression Dichotomy

Every programmer, working within the constraints of a particular operating system and programming language, uses programming techniques to implement a given task or set of tasks in a computer program.²³ These techniques may fall either on the side of expression or ideas and algorithms.²⁴ While the creative choice and ordering of commands, organization of routines and sub-routines, allocation and use of system memory, even the choice of algorithms will be copyrightable, the ideas and algorithms themselves will not be protectible by copyright.

The idea/expression dichotomy is recognized by the EU Software Directive as well as under U.S. copyright law. Section 102(b) of the Copyright Act of 1976²⁵ precludes “any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in [an original] work [of authorship]” from copyright protection. Under European copyright law, Article 1(2) of the EU Software

²¹ In technical terms, interfaces are computer codes “which provide for [...] interconnection and interaction between elements of software and hardware“. See EU Software Directive, *supra* note 10, recital 11, at 43.

²² BAINBRIDGE, *supra* note 4, at 107-8.

²³ Joseph Haaf, *The EC Directive on the Legal Protection of Computer Programs: Decompilation and Security for Confidential Programming Techniques*, 30 COLUM. J. TRANSNAT’L L. 401, 412 (1994).

²⁴ The Merriam-Webster Dictionary broadly defines an algorithm as “a step-by-step procedure for solving a problem or accomplishing some end especially by a computer.” Also the U.S. Supreme Court has given a definition of an algorithm—that is as a “procedure for solving a given type of mathematical problem.” See *Gottschalk v. Benson*, 409 U.S. 63, 65 (1972). The Explanatory Memorandum accompanying the Proposed Directive compared algorithms with the “words by which the poet or the novelist creates his work of literature. See *Commission Preparatory Memorandum of a Proposal for a Council Directive on the legal protection of computer programs*, para. 2.4., 1989 O.J. (C 91) 4 [hereinafter Commission Preparatory Memorandum].

²⁵ 17 U.S.C. § 102(b) (1990).

Directive provides similarly that “ideas and principles which underlie any element of a computer program, including those which underlie its interfaces, are not protected by copyright under this Directive.”²⁶

Interfaces in particular are of a hybrid character as they consist of an interwoven structure of both copyrightable and utilitarian (functional) elements.²⁷ On the one hand, the expression requirement may be satisfied by the selection that is made of the various solutions and the ways in which these steps (i.e. algorithms) are ultimately expressed. On the other hand, these particular characteristics implement non-protectible functional features such as speed and efficiency of the target program.²⁸

Notwithstanding the codification of the idea/expression dichotomy in both the United States and at European level, neither of both jurisdictions defines the conclusive criterion to distinguish between idea and expression embedded in the same interface. Assumably it will ultimately lie with the courts to draw the line between protected and non-protected elements of interfaces. Likely the crucial question of delineating between idea and expression of an interface will turn on the criterion of originality.²⁹

1.3.2. The Originality Standard

1.3.2.1. Different Levels of Originality in Europe

Currently, European copyright law is administered by twenty seven separate national laws—all are said to harmonize to a minimum extent in compliance with a bundle of EU Directives.³⁰ In spite of its intended harmonizing effect, the Council Directive on the legal protection of computer programs, however, does not explicitly dictate whether computer code in general nor interfaces in particular are

²⁶ EU Software Directive, *supra* note 10, art. 1, at 44.

²⁷ The line demarcating programming techniques as expression and function is not clear at all. *See, e.g.*, EU Software Directive, *supra* note 10, recital 14, at 43, reading that “*to the extent* that logic, algorithms and programming languages comprise ideas and principles, those ideas and principles are not protected under the Directive” (emphasis added).

²⁸ *See* Commission Preparatory Memorandum, *supra* note 24, para. 2.3., at 5.

²⁹ JOSEF DREXL, WHAT IS PROTECTED IN A COMPUTER PROGRAM?: COPYRIGHT PROTECTION IN THE UNITED STATES AND EUROPE (1994) 98-9.

³⁰ In 1991, the Directive on the Legal Protection of Computer Programs was the first of five Directives adopted by the European legislators with the objective of harmonizing the copyright laws in the then-13 Member States. *See* Gaster, *supra* note 9, at 15.

ideas or protected expression.³¹ In both jurisdictions the rule has evolved that copying of and translating the principles and methods of operation (i.e., interface specifications) contained in the original computer program and their subsequent implementation in the software programmer's own independently created computer program does not constitute copyright infringement.³² However, the scope of the rule is vague in either copyright law system and thus leaves software programmers in a legal vacuum.

Also the Berne Convention³³ lacks the guidance as to what constitutes "originality" in terms of quantity and quality. Consequently, differences in the copyright laws are still abound. These differences, above all, arise between the common law countries (i.e., the United Kingdom and Ireland) on the one hand, and the continental, civil law, countries on the other. Essentially, there are two mainstream interpretation of the originality concept: the German standard of originality and the U.K. interpretation of originality.

Under German copyright law, only an author's own intellectual creations are eligible to copyright protection. In the case *Inkassoprogramm*³⁴ the Federal Supreme Court applied the so-called "average programmer-test" to determine whether or not a computer program constitutes an own intellectual creation. The court created a two-step test. First, the individuality required for computer code to be protected is scrutinized. For it the code in question will be compared to similar codes in order to gauge its distinction in terms of selection, collection, arrangement and division of information. A second step requires the program in question to demonstrate that it reaches beyond the skills of an average software programmer.

³¹ The former Article 1(4)(a) of the Proposed Directive merely called for the same conditions as regards originality as apply to other (traditional) literary works. See *Commission Proposal for a Council Directive on the legal protection of computer programs*, art. 1, 1989 O.J. (C 91) 13 [hereinafter Proposed Directive].

³² See Thomas C. Vinje, *Die EG-Richtlinie zum Schutz von Computerprogrammen und die Frage der Interoperabilität* [The EC Directive on the Protection of Computer Programs and the Question of Interoperability], 4 GRUR INT. 250, 259 (1992).

³³ Berne Convention for the Protection of Literary and Artistic Works, signed in Paris, Sept. 9, 1886, as revised in Paris, July 24, 1971, S. TREATY DOC. NO. 27, 99th Cong., 2d Sess. (1986) [hereinafter Berne Convention].

³⁴ Bundesgerichtshof [BGH] [Federal Court of Justice] May 9, 1985 (I ZR 52/83), *Entscheidungen des Bundesgerichtshofes in Zivilsachen* [BGHZ] [Federal Supreme Court] 94, 276 (F.R.G.), translated in 17 IIC 1986, 681.

Failing in only one step will result in non-copyrightability of the computer code.³⁵ In short, the average programmer-test is based upon the idea that copyrightable software had to meet a high level of creativity (“*Schöpfungshöhe*”).³⁶ The U.K. approach is understood to require a lesser amount of personal creativity. In British case-law³⁷ the term “original” has been interpreted as relating essentially to origin rather than to substantive considerations such as creativity or novelty. Accordingly, a work will usually be original provided merely that it originates from the author or creator and has not been copied off an already existing work.³⁸ Unlike the German approach, a computer code is considered to be original if it is the fruit of a software developer’s personal labor or even of an automated process.³⁹ As long as the computer code is not itself a copy from another code, the originality standard will be satisfied with ease under U.K. copyright law.

1.3.2.2. Originality under U.S. Copyright Law

The U.S. Copyright Act is silent with respect to classifying computer code as original expression. However, U.S. jurisprudence has unanimously protected code as expression.⁴⁰ In so holding, a tendency in favor of the lower originality threshold of the United Kingdom has been shown to apply in the United States.⁴¹ In addition to the prerequisite that the computer code must not be a copy the code must not be trivial.⁴² In the case *Feist Publications, Inc. v. Rural Telephone Service Co., Inc.*,⁴³ the Supreme Court ruled that originality “necessitates independent creation plus a

³⁵ Thomas Hoeren, *The EC Directive on software protection – a German comment*, July/August 1991 *Computer Law & Practice*, 246; see also JONGEN & MELBOOM, *supra* note 6, at 76-7.

³⁶ *Commission Report to the Council, the European Parliament and the Economic and Social Committee on the implementation and effects of Directive 91/250/EEC on the legal protection of computer programs*, at 8, COM (2000) 199 final (Apr. 10, 2000) [hereinafter *Commission Report*].

³⁷ See *University of London Press v. Universal Tutorial Press*, (1916) 2 Ch. 601 (U.K.) (holding that the originality threshold “does not require that the expression must be in an original or novel form, but that the work must not be copied from another work – that it should originate from the author”).

³⁸ MICHEL M. WALTER, *EUROPÄISCHES URHEBERRECHT* [European Copyright Law] (2001) 123.

³⁹ Jaap H. Spoor, *Copyright Protection and Reverse Engineering of Software: Implementation and Effects of the EC Directive*, 19 U. DAYTON L. REV. 1063, 1065-6 (1994).

⁴⁰ See, e.g., *Sony Computer Entertainment, Inc. v. Connectix Corp.*, 203 F.3d 596, 599 (9th Cir. 2000) [hereinafter *Sony*]; *Whelan Assocs., Inc. v. Jaslow Dental Laboratory, Inc.*, 797 F.2d 1222, 1236 (3d Cir. 1986); *Morrissey v. Proctor & Gamble Co.*, 379 F.2d 675, 678-9 (1st Cir. 1967).

⁴¹ Cf. *Alfred Bell & Co. v. Catalada Fine Arts Inc.*, 191 F.2d 99, 102-3 (2nd Cir. 1951). (holding that independent creation satisfies the threshold originality requirement).

⁴² Hoeren, *supra* note 35, at 247; WALTER, *supra* note 38, at 125.

⁴³ 499 U.S. 340 (1991) [hereinafter *Feist*] (discarding industrious effort, the so-called “sweat of the brow” doctrine, as a standard for copyrightability).

modicum of creativity.” The Court held that a compilation of information is not protected by copyright if it is obvious and commonplace; i.e. a work merely having required some effort *per se* does not trigger off copyright protection.⁴⁴ That means that qualitative criteria should not be applied to copyright “outside of the narrowest and most obvious limit.”⁴⁵ In conclusion, the U.S. approach appears to comport rather with the low U.K. threshold than with the strong German standard.

1.3.2.3. The Originality Standard of the EU Software Directive

The Commission Green Paper suggested a reduced standard of originality in accordance with the Anglo-American copyright concept⁴⁶ serving as a model.⁴⁷ This concept was finally adopted in the final draft of the Directive. Article 1(3) states “[a] computer program shall be protected if it is original in the sense that it is the author’s own intellectual creation [and] [n]o other criteria shall be applied to determine its eligibility for protection.”⁴⁸ Furthermore, the preamble of the Directive expresses that no qualitative or aesthetic test of merit shall be applied to determine whether a computer program is an original work.⁴⁹ Although the preamble explicitly mentions the “investment of considerable human, technical and financial resources”⁵⁰ required for developing a computer program, the mere amount of labor does not, by itself, reach the originality threshold for copyright protection under the legal provision of the Directive.⁵¹ Thus the “sweat of the brow” doctrine, like in the United States, cannot be said to be accepted under European copyright law.

⁴⁴ *Feist*, 499 U.S. at 340.

⁴⁵ See *Bleistein v. Donaldson Lithographing Co.*, 188 U.S. 239, 251 (1903).

⁴⁶ The “Anglo-American” originality concept refers to the low threshold test for originality of copyrightable computer programs as applied under U.K. and U.S. copyright laws.

⁴⁷ *Commission Green Paper on Copyright and the Challenge of Technology – Copyright Issues requiring Immediate Action*, para. 5.6.3., at 187, COM (1988) 172 final (June 7, 1988) [hereinafter *Commission Green Paper*] (regarding a computer program to be “original” if it is to manifest the result of the creator’s own intellectual effort and is not itself a copy).

⁴⁸ EU Software Directive, *supra* note 10, art. 1(3), at 44.

⁴⁹ *Id.*, recital 8, at 42. However, some Member States, like Austria and Italy, have not implemented the wording of the Preamble by excluding any requirement of aesthetics and quality.

⁵⁰ *Id.*, recital 2, at 42 reads in full text:

“Whereas the development of computer programs requires investment of considerable human, technical and financial resources while computer programs can be copied at a fraction of the cost needed to develop them independently.”

⁵¹ DREXL, *supra* note 29, at 98.

There may be compelling reasons why the European legislators preferred to introduce an Anglo-American-like standard rather than the more demanding German interpretation of originality. The reluctance to grant copyright protection for computer code was widely criticized in national and international literature by pointing out that the narrow interpretation adopted in the German *Inkassoprogramm* case would lead to only a small minority of all computer programs to be protected against piracy.⁵² At the same time, adopting the German standard would necessarily result in holding standard programs, or certain program routines and modules not to be protectible.⁵³ The broader interpretation, moreover, finds further justification in the unique nature of computer code.⁵⁴ A lower threshold of qualitative criteria for copyright protection of computer code seems appropriate to comply with the fact that computer programs may be of tremendous commercial value although they lack aesthetic appeal and are only of minimal creativity.⁵⁵ Even if programs are of aesthetic impact, judges and lawyers usually are not competent in making qualitative decisions regarding what copyright protection computer programs should deserve.⁵⁶ Jurisdictions having been adhering to more demanding threshold requirements so far will have to accept the lower standard set forth in the Directive. Germany, for example, has already complied with the European standard⁵⁷ but not without criticizing that new trends and methods in software development, such as standardization, object-oriented

⁵² Cf. BRIDGET CZARNOTA & ROBERT J. HART, LEGAL PROTECTION OF COMPUTER PROGRAMS IN EUROPE: A GUIDE TO THE EC DIRECTIVE (1993) 9; Hoeren, *supra* note 35, at 246; JONGEN & MEIJBOOM, *supra* note 6, at 79.

⁵³ See Spoor, *supra* note 39, at 1066.

⁵⁴ It would seem that given their relatively short length or their simplicity, most interface implementations would not meet a sophisticated originality threshold in order to be protected under copyright as such. See Thomas Dreier, *The Council Directive of May 14 1991 on the Legal Protection of Computer Programs*, at 325 (1991); see also Vinje, *supra* note 32, at 259-60.

⁵⁵ See JONGEN & MEIJBOOM, *supra* note 6, at 220; see also Vinje, *supra* note 32, at 259 (stating that interface implementations because of their short length and arbitrariness do not reach the originality threshold for copyright protection).

⁵⁶ Thus, the EU Software Directive appears to embrace the *Bleistein v. Donaldson Lithographing Co.* nondiscrimination theory found in U.S. copyright law. See *Bleistein v. Donaldson Lithographing Co.*, 188 U.S. 239, 252-3 (1903).

⁵⁷ For example, the German Bundesgerichtshof (BGH) [Federal Court of Justice] overruled its precedent decision in the *Inkassoprogramm* case while expressly referring to the EU Software Directive. See *Entscheidungen des Bundesgerichtshofes* [BGH CR] 1993, 752 – *Buchhaltungsprogramm* (F.R.G.), discussed in GRUR 1994, 39; BGH CR 1994, 275 – *Holzhandlungsprogramm* (F.R.G.), discussed in GRUR 1994, 363.

Also the Commission noted that “the German eligibility criterion has thus been brought in line with the Directive’s terms.” See Commission Report, *infra* note 36, at 8.

programming and the use of automated tools, will lead to computer code being less and less original.⁵⁸ On the other side of the spectrum, it is not yet clear whether the United Kingdom will have to raise their originality threshold in order to fully abide with the Directive.⁵⁹

By referring to “the author’s own intellectual creation” the Directive leaves room for different interpretations in different EU Member States.⁶⁰ Thus, failure to adopt a uniform European originality standard has the unfortunate effect of perpetuating the significant differences between the EU Member States themselves and vis-à-vis the United States, thereby affecting the trade of computer software within the EU as well as beyond European borders.⁶¹

As for European copyright law, it lies in the purview of the ECJ to closely examine if and to what extent the Member States have implemented the EU Software Directive, and, if necessary, to bring these national implementations into line⁶² via Article 1(3)⁶³ as the basis for a liberal interpretation of the originality standard in Europe.

2. Decompilation and Intermediate Copying

Bearing in mind that ideas and techniques are usually less accessible in computer programs than in other forms of copyright, it seems only proper that this imbalance should be reconsidered.⁶⁴ In fact, despite the complex nature of computer code, the

⁵⁸ Spoor, *supra* note 39, at 1067.

⁵⁹ The United Kingdom has not yet implemented the originality standard of the Directive. Given the traditionally low level of originality in the United Kingdom it might be questionable whether trivial or even non-original programs will still be protected under national copyright law. See Gaster, *supra* note 9, at 14; see also DREXL, *supra* note 29, at 98.

⁶⁰ Spoor, *supra* note 39, at 1074.

⁶¹ CZARNOTA & HART, *supra* note 52, at 9, 43.

⁶² See Mindy J. Weichselbaum, *The EEC Directive on the Legal Protection of Computer Programs and U.S. Copyright Law: Should Copyright Law Permit Reverse Engineering of Computer Programs?*, 14 FORDHAM INT’L L.J. 1027, 1067 n. 299 (1991); see also Spoor, *supra* note 39, at 1071-2 (assuming that it will take years for the ECJ to bring the various national interpretations into line).

⁶³ Vinje, *supra* note 32, at 260.

⁶⁴ BAINBRIDGE, *supra* note 4, at 108.

ideas and techniques underlying the creation of a computer code are not beyond discovery.⁶⁵

Decompilation is the process by which ideas and techniques as functional elements embodied in computer program code are made available by means of an operation.⁶⁶ In technical terms, software programmers may undergo different operations of how to gain access to the source code, such as decompilation, disassembly, black-box analysis or a “clean room” procedure.

In redressing the natural imbalance, software programmers use devices called “decompilers”⁶⁷ or “disassemblers”⁶⁸ to read the “zeros” and “ones” that are produced while the program is run. The resulting object code version is stored in computer memory. Lastly, the object code is translated into source code.⁶⁹ The definition of translation done in the reverse process of decompilation and disassembly refers to conversion meaning that the original target program code must be used as a model for the new program, it must at least be referred to by the reverse engineer.⁷⁰ The product as a result of decompilation or disassembly is a rough and ready *doppelgänger* of the original source code, providing an understanding of, at least, much if not all of the structure and operation of the decompiled computer program.⁷¹ It is either procedure and not the final product of reverse engineering that is questionable in legal terms.⁷² Normally, these reverse processes involve copying of the copyrighted target computer code.⁷³

⁶⁵ In fact, reverse engineering (including decompiling or disassembling) copyrighted computer programs is used extensively in the industry. *See, e.g.,* Robert V. Donahoe, *Does Intermediate Copying of Computer Software for the Purpose of Reverse Engineering a Non-Infringing Product Infringe the Copyright in the Software?*, 2001 B.C. INTEL. PROP. & TECH. F. 111301, n.16 (2001).

⁶⁶ BAINBRIDGE, *supra* note 4, at 113.

⁶⁷ Decompiling a computer program to retrieve the original high-level language source code requires knowledge of the exact version of high-level language. *Id.* at 109.

⁶⁸ Disassembly unlocks the ideas and techniques embedded in the object code version of a program and produces assembly language from the object code version of the target program. *Id.*

⁶⁹ WALTER, *supra* note 38, at 210-1, 232 et seq.

⁷⁰ BAINBRIDGE, *supra* note 4, at 109.

⁷¹ NIMMER ON COPYRIGHT, *supra* note 15, § 13.05[D][4], at 13-230.18; *see also E.F. Johnson Co. v. Uniden Corp.*, 623 F. Supp. 1485, 1488-90 (D. Minn. 1985) (indicating that decompilation does not yield a perfect copy of the source code, but can afford an understanding of much of the structure and operation of a program).

⁷² Johnson-Laird, *supra* note 7, at 843 (citing cases in which the legal status of reverse engineering is conditional upon the process of decompilation itself rather than the resulting product).

⁷³ Donahoe, *supra* note 65, at 111301.

Under British copyright law, decompilation and disassembly in relation to a computer program constitutes a translation which may be subsumed under the exclusive right of making an adaptation.⁷⁴ In contrast to the U.K. approach, the German understanding of making an adaptation would require the compiler or assembler to invest creative efforts what, however, is not needed when decompiling or disassembling object code into source code.⁷⁵ The problem with a meaning of translation and adaptation in respect of computer code wide enough to cover acts of decompilation and disassembly occurs in the context of the idea/expression dichotomy. Rewriting a computer program in a different high-level language involves going back to the basic ideas and techniques rather than making a line for line translation⁷⁶ since the software programmer is not particularly concerned with converting the actual code of the original target program.⁷⁷ Thus, protecting acts of decompilation and disassembly as translation or adaptation would be overreaching to the extent of the uncopyrightable ideas and principles contained in the original target program code.⁷⁸

After all, the question is whether those acts qualify for infringement of the software copyright owner's exclusive right to reproduction, translation and/or making an adaptation.⁷⁹

⁷⁴ Section 21 of the Copyright, Designs and Patents Act of 1988 (U.K.):

“[Making an adaptation includes] a version of the program in which it is converted into or out of a computer language or code or into a different computer language or code, otherwise than incidentally in the course of running the program.”

⁷⁵ Cf. MORITZ & TYBUSSECK, *supra* note 16, at 71-2 (in describing the creation of a computer program the authors insist on the outcome regardless of whether there is 100 per cent identity of the object code “copy” with the source code “original”).

⁷⁶ The transformation of object code to source code does not result in the original source code identical to the one embodied in the original target code. The decompiled source code, therefore, is often called “pseudo-source code”. See HANNS ULLRICH & EBERHARD KÖRNER, *DER INTERNATIONALE SOFTWAREVERTRAG* [The International Software Contract] (1995) 77 n.150.

⁷⁷ BAINBRIDGE, *supra* note 4, at 110.

⁷⁸ J. F. Verstrynge confirmed in a speech held at World Computer Law Congress in 1991 in Los Angeles that acts of translation from one language into another—e.g., from Pascal to C—and adaptation of the target program code are not subjected to the exception of Article 6 and thus constitute copyright infringement. Cf. Verstrynge, J. F., *Protecting Intellectual Property Rights within the New Pan-European Framework – Computer Software*, Paper presented at the World Computer Law Congress 1991 in Westin Bonaventura, Los Angeles (Apr. 18-28, 1991), at 10-4.

⁷⁹ 17. U.S.C. § 106(1) and (2) (2002) saying that “the owner of copyright under this title has the exclusive rights to do and to authorize any of the following: (1) to reproduce the copyrighted work in copies or phonorecords; (2) to prepare derivative works based upon the copyrighted work;”

See Article 4 of the EU Software Directive, as the pertinent provision in the EU copyright law, reads as follows: “[T]he exclusive rights of the rightholder within the meaning of Article 2, shall include the right to do or to authorize: (a) the permanent or temporary reproduction of a computer

The answer necessitates to take notice of computer programs as appearing in different manifestations (e.g. object code, source code, screen displays), to wit, in functional, i.e. unprotected, as well as in artistic, i.e. protectible, manifestations. Consequently, in the process of gaining access to those utilitarian elements of a copyrighted program—say for the purpose of achieving interoperability with the decompiled copyrighted product—software programmers will hit on both copyrighted expression and non-copyrightable, functional elements of the target computer code. There are to distinguish ideas and functional concepts underlying one type of computer programs, such as word processing and spreadsheets, that are readily discernible without need for an intermediate (preliminary) step, for example, in form of decompilation because the operation of such programs is visible on the computer screen. And there are ideas and functional elements of another type of computer programs, such as operating system software, that are not visible to the user when operating. In this case, the crux is that it is impossible to distinguish the protected from the unprotected elements of a copyrighted computer program unless the object code is decompiled or disassembled.⁸⁰ In doing so, the computer copies the target program code into its memory (e.g. hard disc, ROM, RAM), transforms the program and saves the transformed source code back into its memory or produces a paper copy of it.⁸¹ Here, the interface between making a copy of a copyrighted computer code in the course of decompiling computer code and infringement of the software copyright owner's exclusive reproduction right becomes visible. Thus, certainly the permissibility of intermediate copies made during the process of decompiling or disassembling the target code is in question. The lawfulness of these reverse processes under each copyright system depends on how the technical fact that the copying is done only for a limited period of time as an intermediate step is weighed in legal terms.

program by any means and in any form, in part or in whole. Insofar as loading, displaying, running, transmission or storage of the computer program necessitate such reproduction, such acts shall be subject to authorization by the rightholder; (b) the translation, adaptation, arrangement and any other alteration of a computer program and the reproduction of the results thereof , without prejudice to the rights of the person who alters the program[.]”

The exclusive rights provided to the copyright owner under the EU Software Directive closely mimic the rights granted under the U.S. Copyright Act.

⁸⁰ *Cf. Sega*, 977 F.2d at 1520.

⁸¹ Joe L. Gage, Jr., *Copyright Law and Reverse Engineering: Have Recent Decisions Taken the Fair Out of Use?*, 46 BAYLOR L. REV. 183 (1994).

The EU Software Directive explicitly identifies “loading, displaying, running, transmission or storage of the computer program” to require potentially infringing reproduction.⁸² The same is true under U.S. case law.⁸³ In the U.S. case *MAI Systems Corp. v. Peak Computer, Inc.*,⁸⁴ the principle has been established that making an infringing copy of a computer program⁸⁵ does not necessarily require the making of a hard copy, a translated copy or another disk copy of the original target code.⁸⁶ Given that computer code (in whatever form) is a protectible form of expression,⁸⁷ reverse engineers at both sides of the Atlantic infringe directly the author’s exclusive right of reproduction by creating an intermediate copy. Therefore, under both European and U.S. copyright law, technically, making a copy or adaptation or both of the reconstructed source code, whether only a single one, whether merely a preliminary one to further uses rather than a final one, will constitute infringement of the copyrighted computer program.

In light of a computer program’s technical characteristics, some might argue whether it is in fact the reverse process itself which is the infringing conduct rather than the end product of the reverse process. Intermediate copies are an indispensable by-product of decompiling computer software. Once the code is transformed, the reverse engineer can analyze the structure of the target program and put the information to a variety of uses. To put it in other words, these copies are referred to as “intermediate” copies because the protected expression contained

⁸² EU Software Directive, *supra* note 10, art. 4(a), at 44 (enumerating as restricted acts, unless authorized by the rightholder, “permanent or temporary reproduction of a computer program by any means and in any form, in part or in whole [including] ... loading, displaying, running, transmission or storage of the computer program [that] necessitate such reproduction”).

⁸³ NIMMER ON COPYRIGHT, *supra* note 15, § 13.05[D][4] at 13-230.

⁸⁴ 991 F.2d 511, 519 (9th Cir. 1993).

⁸⁵ 17 U.S.C. § 106(1) (granting the copyright owner the exclusive right “to reproduce the work in copies” and § 501 providing that “[a]nyone who violates any of the exclusive rights of the copyright owner as provided by sections 106 through 118 ... is an infringer of the copyright”).

⁸⁶ Any intermediate copy of the object code made in order for a programmer to read and comprehend it suffices to meet the fixation threshold of the Copyright Act. Johnson-Laird, *supra* note 7, at 892.

⁸⁷ See *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1249 (3d Cir. 1983) (“[A] computer program, whether in object code or source code, is a ‘literary work’ and is protected from unauthorized copying, whether from its object or source code version.”); see EU Software Directive, *supra* note 10, recital 20, (“[w]hereas the unauthorized reproduction, translation, adaptation or transformation of the form of the code in which a copy of a computer program has been made available constitutes an infringement of the exclusive rights of the author[.]”).

therein does not necessarily appear in the reverse engineered product.⁸⁸ Indeed, the most common product of reverse engineering⁸⁹ is elements of decompiled information which will be recoded and integrated into the new independently created program.⁹⁰ Against this backdrop, holding intermediate copyists liable for copyright infringement in the course of producing an otherwise non-infringing computer program may seem counterintuitive or simply unfair.

Henceforth, in determining the legitimacy of decompiling a copyrighted computer software both U.S. courts and the drafters of the EU Software Directive reflected to the unconventionally copyrightable nature of computer code in the light of copyright policy and fairness considerations.⁹¹

3. Copyright Policy Considerations of Decompiling Computer Software

3.1. A Challenge For Traditional Copyright Policy

3.1.1. Innovation and Consumer Welfare

The cross-border doctrinal debate⁹² on decompilation of computer software centers on the tension between the fundamental rights to intellectual property and the ultimate copyright objective “to stimulate artistic creativity for the general public

⁸⁸ See Donahoe, *supra* note 65, at 111301; see also *Sony Computer Entertainment, Inc. v. Connectix Corp.*, 203 F.3d 596, 600 (9th Cir. 2000) [hereinafter Sony].

⁸⁹ In *Kewanee Oil Co. v. Bicron Corp.*, 416 U.S. 470, 476 (1974), the U.S. Supreme Court defined “reverse engineering” as “starting with the known product and working backward to divine the process which aided in its development or manufacture.” See also BLACK’S LAW DICTIONARY 1345 (8th ed. 2004) (defining “reverse engineering” as “the process of discovering how an invention works by inspecting and studying it, esp[ecially] by taking it apart in order to learn how it works and how to copy it and improve it”).

Although the terminology of “reverse engineering” and “decompilation” is used—probably mistakenly—interchangeably, in fact decompiling needs to be distinguished from reverse engineering, in that reverse engineering goes beyond decompilation by including rebuilding the decompiled computer program on the basis of the result of the decompilation.

⁹⁰ Verstrynge, *supra* note 78, at 10-4.

⁹¹ See GOLDSTEIN ON COPYRIGHT, *supra* note 12, § 7.2.1.4.c, at 7:51; see generally Terril Lewis, *Reverse Engineering of Software: An Assessment of the Legality of Intermediate Copying*, 20 LOY. L.A. ENT. L. REV. 561, 563-4 (2000).

⁹² For a discussion on how the doctrinal debate was articulated by the various lobbying groups in the course of drafting the EU Software Directive and the DMCA see *infra* Part III.3. and Part V.1.1.

good.”⁹³ As shown above, intermediate copying constitutes a *prima facie* infringement of the copyright owner’s exclusive reproduction right. By prohibiting intermediate copying the fundamental goal of copyright policy to devote copyrighted works to their socially most beneficial use, through the distribution of know how and knowledge would be undermined.⁹⁴ In a broader context, prohibiting decompilation in the process of reverse engineering will stifle innovation.⁹⁵ Even when reverse engineering does not lead to additional innovation, it would still promote consumer welfare by providing consumers with a competing product at a lower price.⁹⁶

However, flipping the coin and simply permitting intermediate copying will result in a violation of a copyright owner’s fundamental right to intellectual property.⁹⁷ The inability to develop compatible products would seriously disrupt the computer industry and greatly increase the development and research (R&D) expenses of computer products. In a time where there are a great multitude of hardware and software manufacturers, the compatibility of software with other software and across the many computer platforms that are available is integral to the effective and efficient use of computers.⁹⁸ A right to decompilation for the purpose of accomplishing interoperability of the reverse engineered program, moreover, can increase competition in the marketplace, lead to lower prices and spur follow-on innovations by secondcomers⁹⁹ which ultimately will increase consumer choices.¹⁰⁰ Furthermore, disclosure of APIs will drive demand for the platform since new and perhaps improved applications will be available for the platform whereby demand for the platform itself will skyrocket. For example, IBM required Microsoft to

⁹³ See, e.g., *Sony Corporation of America v. Universal City Studios, Inc.*, 464 U.S. 417, 432 (1984).

⁹⁴ See *Harper & Row Publishers, Inc. v. Nation Enterprises*, 471 U.S. 539, 545-6 (1985).

⁹⁵ Samuelson & Scotchmer, *supra* note 5, at 1583.

⁹⁶ *Id.*, at 1582.

⁹⁷ See Ivan Rothman, *From Sega to Sony and Beyond: An Alternative Legal Basis for Software Reverse Engineering*, 18 NO. 3 INTELL. PROP. L. NEWSL. 1 (2000).

⁹⁸ See Derek Prestin, *Where To Draw The Line Between Reverse Engineering And Infringement: Sony Computer Entertainment, Inc. v. Connectix Corp.*, 3 MINN. INTELL. PROP. REV. 137, 173 (2002).

⁹⁹ Samuelson & Scotchmer, *supra* note 5, at 1588.

¹⁰⁰ Carla Meninsky, *Locked Out: New Hazards of Reverse Engineering*, 21 J. MARSHALL J. COMPUTER & INFO. L. 591, 592 (2003) (without the ability to create new products compatible with each other consumers will be harmed by having limited applications deriving from a single source from which to choose).

license the APIs to its operating system to enable application developers to write programs for the IBM PC. This resulted in “[a] large library of off-the-shelf IBM PC compatible application software (particularly Lotus 1-2-3) [that] made the IBM PC an attractive platform.”¹⁰¹ Due to disclosure of Microsoft’s interface specifications in order to incentivise the development of IBM PC compatible user programs and in connection therewith the increase of demand for IBM PCs the IBM PC rapidly achieved a substantial foothold in the market.¹⁰²

To summarize, the classic concern in intellectual property law in general is the tradeoff between private and public interests in software creation. The resulting challenge for law is to regulate the copyright in software on the one hand as to incentivise the production of computer programs and in so doing copyright protection should not be so inhibitive as to prevent subsequent innovations in software production on the other hand. What law, either copyright law or competition law, might be the right platform to weigh those interests is another question and will be discussed elsewhere in this paper.¹⁰³

3.1.2. Exclusive Copyrights

In effect, a prohibition on decompilation would give software producers perpetual exclusive rights in unpatented interfaces.¹⁰⁴ Otherwise legalizing software decompilation would prevent authors of computer programs from enforcing their exclusive rights in their works. However, on balance, a legal rule favoring decompilation and subsequent reverse engineering is economically sound since the original software programmers is nevertheless protected against secondcomers in two ways. First, R&D costs of the entire process of reverse engineering may dissuade secondcomers at all. Second, lead time due to difficulties of reverse engineering ensures the monopoly of the original software programmer so long as the secondcomer is engaged in reverse engineering the original program. In other

¹⁰¹ Samuelson & Scotchmer, *supra* note 5, at 1616.

¹⁰² *Id.*

¹⁰³ For a discussion see Part VII.2.2.

¹⁰⁴ Samuelson & Scotchmer, *supra* note 5, at 1590.

words, lead time serves the same function as a short-lived intellectual property right.¹⁰⁵

3.2. *Purposes of Decompiling Computer Programs*

What makes such a technique as decompilation contentious is that it technically infringes on the reproduction rights of the copyright owner of the underlying computer program, regardless of its ultimate purpose or application. Thus the question goes of whether or not the purpose of decompilation shall determine the legal quality of intermediate copying.

Decompilation for reverse engineering ends may have several purposes, either commercial or non-commercial: diagnosis (i.e., research and error correction), interoperability, and competition.¹⁰⁶ The knowledge obtained through the reverse process may be used for research and education, which basically does not involve creating a subsequent work. Besides pirating a computer program's protected expression, which will certainly be regarded as an infringing act, a competitor may legally copy a computer code for the purpose of determining the program's unprotected interface specifications in order to produce non-infringing, though competing, programs that operate on computers running the copyrighted program.¹⁰⁷ The compatible product may be intended to interact with either the underlying program or the platform upon which it operates. Lastly, intermediate copying may result in designing a competing program which may serve as a direct substitute for the underlying program. A defendant competitor who copies or adapts computer code through acts of decompilation will most likely meet with antipathy, yet especially, if the reverse engineered program based on the

¹⁰⁵ *Id.*, at 1582, 1586. However, a reverse engineer nonetheless is likely to spend less time and money to discern the know how underlying in the original target program than the initial software programmer spent in developing its program. *See id.* at 1587.

¹⁰⁶ *See* HABERSTUMPF in LEHMANN, *supra* note 2, at 159; *see also* Samuelson & Scotchmer, *supra* note 5, at 1582 (indicating that there may be diverse reasons for reverse engineering, with the purpose of making a competing product as the most common because most economically significant reason to reverse-engineer in the industrial context); *see also* Johnson-Laird, *supra* note 7, at 846.

¹⁰⁷ *See id.*

information decompiled reaches the market.¹⁰⁸ Naturally, that proves the most controversial use of interface information discovered through decompilation.¹⁰⁹

3.3. *Is Article 6 Fair to Common Law Traditions in Europe?*

As the paper will revert to in great detail later, most of the EU Member States and the United States (together with the common law EU Member States that is the United Kingdom and Ireland) differ on the positions they have taken with regard to each one of legalizing software decompilation. U.S. case law thus far has resorted to the flexible fair use doctrine on a case-by-case basis. The European legislators took a more rigorous approach by allowing decompilation under certain circumstances *ex lege*.

In the course of legislative drafting at European level three approaches were considered: (i) either to abstain from drawing up a decompilation provision in the EU Software Directive and to leave the issue with the jurisprudence of the European Communities, (ii) to introduce a fair use concept as known in common law legal traditions, or (iii) to include an express provision allowing decompilation.¹¹⁰ A fair use exception would have provided for that certain circumstances would justify the performance of otherwise restricted acts, regardless of whether other means would exist to accomplish the same objective. For example, “fair dealing” in the United Kingdom or “fair use” in the United States permits new circumstances to be introduced into the scheme of copyright protection and its exceptions in order to be able to uphold the protection for decompilation. Despite the flexibility of a fair use conception that would permit new circumstances, as occurring in the digital world quite frequently, to be introduced into the categories covered by existing exceptions,¹¹¹ such approach would fit uneasily in civil law traditions of most of the EU Member States.

¹⁰⁸ GOLDSTEIN ON COPYRIGHT, *supra* note 12, § 7.2.1.4.c, at 7:51.

¹⁰⁹ Celine M. Guillou, *The Reverse Engineering of Computer Software in Europe and the United States: A Comparative Approach*, 22 COLUM.-VLA J.L. & ARTS 533 (1998).

¹¹⁰ WALTER, *supra* note 38, at 209.

¹¹¹ See, e.g., CZARNOTA & HART, *supra* note 52, at 75; but see MARLY, *Urheberrechtsschutz für Computersoftware in der Europäischen Union* [Copyright Protection for Computer Software in the European Union] (1995) 294 (stressing the flexible character of the fair dealing doctrine and at the same time pointing out the disadvantage of legal uncertainty as result of the yet not settled case law regarding the doctrine’s mainstream).

Therefore, attempts were made to identify circumstances which the few national courts applying the fair use concept in Europe might find fair if they were applying the fair dealing or fair use doctrine.¹¹² Those circumstances were codified into a series of checks and balances.¹¹³ This approach finally has found its expression in Recital 22 of the EU Software Directive saying that decompiling constitutes a “fair” practice under the conditions provided in the Directive.¹¹⁴

By simply labeling acts of decompilation under particular circumstances as “fair” the European legislator in fact urged common law jurisdictions in Europe to adopt a rigidly devised statute. There was not discussed even the notion of a possible “escape” clause in drafting the EU Software Directive as a means not to curtail the flexible case-by-case definition of fairness under common law traditions.

An explanation for the adoption of Article 6 without any “escape hatch” for common law Member States within the European Communities is found in the overriding goal of the EU Software Directive as expressed in Recital 4 that

certain differences in the legal protection of computer programs offered by the laws of the Member States have direct and negative effects on the functioning of the common market as regards computer programs and such differences could well become greater as Member States introduce new legislation on this subject.¹¹⁵

Despite the EU legislators’ fears, also the European common law Member States have abided with the rather delicate nature of the text.¹¹⁶ It is bizarre that it is the most contentious provision of Article 6 of the whole EU Software Directive which has been implemented to the fullest extent by the EU Member States.¹¹⁷ But the way to go there has been taking debatful years beginning in 1988.

¹¹² While most of the EU nations are parts of the civil law tradition, there are few nations, such as Eire and the United Kingdom, that are parts of the common law tradition. The copyright law of the United Kingdom, for instance, is such that today it is halfway between the system prevailing in the United States and those of certain EU Member States predicated upon a moral rights structure.

¹¹³ CZARNOTA & HART, *supra* note 52, at 74-6.

¹¹⁴ EU Software Directive, *supra* note 10, recital 22, at 43, stating in full text:

“Whereas it has therefore to be considered that in these limited circumstances only, performance of the acts of reproduction and translation by or on behalf of a person having a right to use a copy of the program is legitimate and compatible with fair practice and must therefore be deemed not to require the authorization of the rightholder[.]”

¹¹⁵ EU Software Directive, *supra* note 10, recital 4, at 42.

¹¹⁶ See *Commission Report*, *supra* note 36, at 13.

¹¹⁷ The text of Article 6 is complex and its clauses are interdependent in a way which would make it difficult to transpose into national legislation in alternative language. Each of the numerous

III. Software Decompilation under EU Copyright Law

1. The European Version of the Reverse Engineering Battle

As early as in 1988,¹¹⁸ the Commission addressed in its Green Paper the contentious legal nature of decompiling computer software,¹¹⁹ and was not afraid of raising the question whether copyright protection should apply to access protocols and interface information of computer programs.¹²⁰ The follow-up drafting of a provision became a political battle over how the answer shall look like.

At the time when the EU Software Directive has been debated, the recently enacted legislation in the United Kingdom, France, Germany, Spain and Denmark contained any express provision on reverse engineering in general or on decompilation in particular.¹²¹ Neither did the U.S. jurisprudence at that time.¹²²

key elements has a counter-balancing effect which means that removal or alteration of one part distorts the effect of another.

¹¹⁸ As a result of the increasingly harmful effects of piracy on the market and the urgency of unifying and harmonizing the laws of the Member States, potential alternatives for protecting software were addressed in the 1988 European Community Commission's Green Paper on Copyright and the Challenge of Technology, which solicited comments on a number of key issues regarding software protection, *inter alia*, reverse engineering. See *Commission Green Paper, supra* note 47, at para. 5.5.7. et seq.

¹¹⁹ See *id.* (but by referring to "reverse engineering" instead of "decompiling" the Commission availed itself of an inaccurate expression and which may be a reason why the particular process of decompiling is often too broadly called "reverse engineering"). See also JONGEN & MEIJBOOM, *supra* note 6, at 14-5.

The Directive as finally adopted correctly designated the process referred to in Article 6 as "decompilation", thus avoiding the term "reverse engineering"—which includes rebuilding the decompiled computer program on the basis of the result of the decompilation. The overall process of reverse engineering is not permitted under the Directive.

¹²⁰ WALTER J. JABUREK, DAS NEUE SOFTWARE-URHEBERRECHT [The New Software Copyright Law] (1993) 21.

¹²¹ CZARNOTA & HART, *supra* note 52, at 75.

¹²² Prior to 1992, litigation at district court level in cases involving reverse engineering matters had proven inconclusive. The majority rule was that copies made during reverse engineering manifested an infringement of the copyright holder's exclusive rights. See, e.g., *SAS Inst., Inc. v. S & H Computer Sys., Inc.*, 605 F.Supp. 816 (M.D. Tenn. 1985) (enjoining defendant from distributing a competing software product that it had developed through copying the source code of SAS's program, in violation of its license agreement with SAS; the district court's ruling that the systematically modifications of plaintiff's source code to disguise its true origin that resulted in the making of innumerable copies for unauthorized use constituted copyright infringement is often cited by opponents of reverse engineering); *E.F. Johnson Co. v. Uniden Corp.*, 623 F.Supp. 1485 (D.Minn. 1985) (holding in dicta that „the mere fact that defendant's engineers dumped, flow charted and analyzed plaintiff's codes does not, in or of itself, establish pirating“ but is rather the

In an attempt to align the legislation of the European Communities with the position taken by the United States, the European legislators¹²³ consulted the U.S. Ambassador in the early part of the debate on a provision of decompilation in the EU Software Directive.¹²⁴ The U.S. government indicated that under U.S. copyright law decompilation was not permitted without authorization of the program owner unless excused by the provisions of the fair use doctrine or as a back-up copy under section 117 of the U.S. Copyright Act of 1976. However, it was further identified that case law has then not provided any cases in which fair use had been applied to decompilation.¹²⁵ It thus might not be surprising, that the initial Proposal for the EU Software Directive lacked for a pertinent exception permitting decompilation.¹²⁶ The silence on the permissibility of reverse

standard practice in the industry“; proponents of reverse engineering of computer software usually quote this case). A more sympathetic attitude towards decompilation process, albeit representing the minority position on the issue, was pursued, for example, in *NEC Corp. V. Intel Corp.*, 645 F.Supp. 590 (N.D. Cal. 1989) (holding that the final derivative work was non-infringing but without addressing the legality of the decompiled code which was the basis of the derivative work).

¹²³ The European Commission is composed of representatives from each Member State and is divided into Directorates General of varying size which individually deal with matters of concern to the European Communities. The Commission initiates the legislative process and presents an initial proposal to the Council of Ministers consisting of members who are appointed by their respective national governments. After the Council reaches a consensus on the proposal, it is reviewed, debated, amended if necessary, and written as draft legislation by the European Parliament, which is typically divided into sub-committees. If the Parliament recommends adoption of the draft, it is then returned to the Commission which, in turn, submits the amended proposal to the Council. The Council works to reach a so-called common position or draft form of the proposal it is willing to adopt. This draft is then returned to the Parliament for a second reading. Parliament issues its final recommendations for adopting, rejecting, or amending the common position so that the Council can officially adopt or reject the proposal. Once a Directive is adopted by the European Communities, the Commission ensures proper implementation of the Directive within Member States' national laws.

¹²⁴ Like the U.S. Government, also the Japanese Government was consulted on the treatment of unauthorized decompilation in the course of reverse engineering requesting information on the position in Japan under copyright law. Similarly to the U.S. response, the Japanese Government indicated that the Japanese Copyright Law did not contain any express provision rendering decompilation itself permitted or prohibited. The response further indicated that there was also no express provision in Japanese legislation nor court decision on the decompilation done to gain access to underlying idea and principles.

¹²⁵ See *Dreier*, *supra* note 54, at 113 (identifying that, prior to 1992, there was not a single case which held decompilation as fair use).

¹²⁶ Cf. *Proposed Directive*, *supra* note 31, compared with *Amended Proposal for a Council Directive on the Legal Protection of Computer Programs*, 1991 O.J. (C 320) 22 et seq. [hereinafter *Amended Proposal*]. The failure to provide an express provision on reverse engineering was viewed as a ban on the practice. The express provision authorizing decompilation when necessary for interoperability first appeared in the amendments suggested by the European Parliament. See 1990 O.J. (C 231) 81 [hereinafter *Parliament Amendments*].

For a detailed legislative history of the EU Software Directive see Guillou, *supra* note 109.

engineering exposed the further drafting process to intense lobbying efforts by groups representing diverse interests in the software industry.

*1.1. The United States' Involvement: A Fight For "The Last Bastion of American Technological Superiority"?*¹²⁷

Two prominent groups were involved in the pre-compromise lobbying activities. On one end of the spectrum, the "ultra-protectionists"¹²⁸—organized in the "Software Action Group for Europe" (SAGE)—preferring restrictive provisions for reverse engineering to preserve their current dominance in the European computer market.¹²⁹ SAGE's arguments were that permitting reverse engineering would be a dramatic change from existing law, without being necessary because manuals and other documentation can be used in lieu of reverse engineering methods. Moreover, SAGE referred to the unique nature of software which would make it easy for second comer to translate and copy software wherein the original developer had spent resources to write the software in no proportion to the subsequent reverse engineering. Thus, the lead time normally enjoyed by the original software writer would be reduced so greatly that the original developer no longer would be able to recoup the research expenditure during the period of temporary monopoly provided by the lead time.¹³⁰ According to SAGE's lobbyists, because lenient reverse engineering provisions would discourage the software industry from engaging in research and development, SAGE favored the strict protection found in the initial

¹²⁷ See J. H. Reichman, *Goldstein on Copyright Law: A Realist's Approach to a Technological Age*, 43 STAN. L. REV. 943, 948 (1991) (book review).

¹²⁸ See Boris Rotenberg, *The Legal Regulation of Software Interoperability in the EU*, NY Jean Monnet Working Paper No. 07/2005, at 15, available at <http://www.jeanmonnetprogram.org/papers/05/050701.pdf> (last visited Feb. 22, 2007).

¹²⁹ SAGE representing 80% of the computer industry is led by International Business Machines (IBM) and Digital Equipment Corp. (DEC). Furthermore, the Business Software Alliance (BSA), whose members are, above all, business software market leaders like Microsoft, Lotus Development Corp., and dominant European hardware and software manufacturers, for instance, Philips and Siemens, are represented by SAGE as well.

¹³⁰ The industry leaders seem to be overly concerned with a free rider problem in an effort to conceal their ulterior motive of protecting their lead time advantage by forcing competing second comer to reinvent an "inefficient wheel". Consequently, software developer, whose product has become the industry standard, can create a *de facto* monopoly. See Linda G. Morrison, *The EC Directive on the Legal Protection of Computer Programs: Does It Leave Room for Reverse Engineering Beyond the Need for Interoperability?*, 25 VAND. J. TRANSNAT'L L. 293, 304, 326 (1992).

Proposal.¹³¹ SAGE found support by the United States Trade Representative (USTR). As a matter of U.S. international policy, the reverse engineering of computer software was regarded anticompetitive. Because the USTR deemed decompilation unlawful under U.S. law, thus, it should not be allowed under European law likewise. To put it in other words, the U.S. government apparently feared that an EU Directive condoning decompilation for the purpose of reverse engineering would permit Japanese competitors to pirate software that had been developed in the United States because those U.S. software programs making their way to the EU market would be subject to reverse engineering under a lenient EU Directive.¹³²

1.2. *Europe's Shot For Open Systems*

As counterpart to SAGE, smaller and primarily European—i.e. competing—computer software companies sought broad rights to use reverse engineering as a means to produce devices and computer programs built upon systems and software of the dominant computer companies.¹³³ They organized themselves in the “European Committee for Interoperable Systems” (ECIS).¹³⁴ ECIS argued that established software markets are dominated by industry giants. Software newcomers as well as small and medium sized companies are dependent on producing software compatible with the products of the market leaders in order to get a, or at least a chance of, stake in the relevant computer market,¹³⁵ thereby preserving a sound computer software market and encouraging competition among

¹³¹ *Id.* at 303.

¹³² Guillou, *supra* note 109, at 533 (presenting SAGE’s stance in view of the fears of monopolistic practices by larger manufacturers, should reverse engineering be prohibited, as SAGE regarded such fears misguided and further referred to European antitrust law as the more appropriate field of law to face such fears).

¹³³ JONGEN & MEIJBOOM, *supra* note 6, at 1 (citing the Commission, that the United States and, to a lesser extent, Japanese software industry enjoy an advantage over the European software industry and wished to strengthen the European industry by creating an unambiguous harmonized – and almost uniform – legal treatment of computer software).

¹³⁴ ECIS was led by Fujitsu and represented members including Bull, and Sun Microsystems (the latter representing U.S. interests within ECIS).

¹³⁵ ECIS and industry newcomers feared that strong protection of user interfaces combined with limited reverse engineering and decompilation rights will convert the EU into merely a marketplace for non-Community products manufactured by U.S. industry leaders.

software developers.¹³⁶ Additionally, consumers would benefit from an evolving and improving software industry without inflating prices.¹³⁷ For ECIS proponents, preserving competition and, to a greater extent, their own survival is depended upon interoperability. In line with ECIS's pro-competitive reasoning, the European Parliament likewise encouraged the introduction of a provision in favor of open systems. Its stance favoring international standardization was driven by the concern that without interoperability small European software producers may have a competitive disadvantage as against large European and U.S. computer firms.¹³⁸ Although the Commission questioned acts of decompiling in terms of efficiency-cost considerations as lengthy, costly and inefficient procedure, while it was usually more efficient for the parties concerned to agree on the terms under which the information would be made available,¹³⁹ ECIS's argumentation nonetheless was logical and persuasive to the Commission.¹⁴⁰ After all, in 1991, the Council enacted the Directive on the legal protection of computer programs, containing a right to decompiling computer programs¹⁴¹—the mostly praised “Article 6 compromise”.

2. The White Flag: Article 6 of the EU Software Directive

Despite the extent of controversy over software decompilation the European legislators, ultimately, managed to create a reasonably balanced legal regime of interests and needs of the market leaders in the computer industry, individual

¹³⁶ In their view—contrary to the fears of SAGE—reverse engineering for purposes of compatibility will not destroy the benefits afforded to the first software developer because the industry leader will still enjoy lead time, albeit shortened, as well as the reputation of being the first innovator on the market. *See Morrison, supra* note 56, at 326.

¹³⁷ “Computer Users in Europe” (CUE) was another lobbying fraction in favor of less restrictive copyright protection. CUE was concerned that prohibiting reverse engineering would stunt the development of “open systems”. Moreover, CUE opined that only a right to reverse engineer may hinder dominant U.S. computer companies to convert their commercial positions into legal monopolies. *See id.* at 302-3.

¹³⁸ LEHMANN, *supra* note 2, at 3-4.

¹³⁹ *Commission Preparatory Memorandum, supra* note 24., para. 3.14. et seq., at 8.

¹⁴⁰ *Id.*, para. 1.3., at 5 (“It is essential to create a legal environment which will afford a degree of protection against unauthorized reproduction...if research and investment in computer technology are to continue at a sufficient level to allow the Community to keep pace with other industrialized countries.”).

¹⁴¹ *See* EU Software Directive, *supra* note 10, art. 6, at 45.

software programmers and users, incorporated in Article 6 of the EU Software Directive.¹⁴² The EU legislator went to treat computer software as a black-box. Provided that the inside is original in the sense that it is the author's personal creation, the black-box will enjoy copyright protection under Article 1 of the EU Software Directive.¹⁴³ Every glance into the box is inevitably preceded by, at least interim, copying in the copyright sense, and therefore is forbidden—save with the copyright owner's authorization—which is the essence of Article 4 of the Directive. Only in two instances second comers may have to look into the black-box by way of decompilation: for correcting errors (Article 5(1)), and for achieving interoperability (Article 6).

Unlike the United States, the EU Software Directive does not pursue a fair use approach,¹⁴⁴ but rather addressed the issue of decompilation directly. The idea behind Article 6 is that decompiling computer software¹⁴⁵ generally shall not be permitted, but is authorized in limited circumstances.¹⁴⁶ According to Article 6(1) of the Directive, acts of reproducing the code or transforming the code into whatever level of human readability do not need the authorization of the copyright holder of the program decompiled provided that they are “indispensable to obtain the information necessary to achieve interoperability of an independently created computer program with other programs.”¹⁴⁷ With view to accomplish

¹⁴² See, e.g., Rotenberg, *supra* note 128.

¹⁴³ EU Software Directive, *supra* note 10, art. 1, at 44, reads in full text:

“1. In accordance with the provisions of this Directive, Member States shall protect computer programs by copyright, as literary works within the meaning of the Berne Convention for the Protection of Literary and Artistic Works. For the purpose of this Directive, the term ‘computer programs’ shall include their preparatory design material.

2. Protection in accordance with this Directive shall apply to the expression in any form of a computer program. Ideas and principles which underlie any element of a computer program, including those which underlie its interfaces, are not protected by copyright under this Directive.

3. A computer program shall be protected if it is original in the sense that it is the author's own intellectual creation. No other criteria shall be applied to determine its eligibility for protection.”

¹⁴⁴ For a discussion of the fair use doctrine under U.S. copyright law see *infra* Part IV.2.

¹⁴⁵ The Directive does not use the term “software” but “computer programs”. In practice, both terms are usually used as synonyms. See JABUREK, *supra* note 8, at 18.

¹⁴⁶ JONGEN & MEJIBOOM, *supra* note 6, 14-5.

¹⁴⁷ EU Software Directive, *supra* note 10, art. 6(1), at 45, reads as follows:

“1. The authorization of the rightholder shall not be required where reproduction of the code and translation of its form within the meaning of Article 4(a) and (b) are indispensable to obtain the information necessary to achieve the interoperability of an independently created computer program with other programs, provided that the following conditions are met:

(a) these acts are performed by the licensee or by another person having a right to use a copy of a program, or on their behalf by a person authorized to do so;

interoperability—to whatever extent—the original code may be reproduced as many times as is necessary to arrive at a form which is sufficiently comprehensible to the decompiling software engineer.¹⁴⁸ Besides reproducing and transforming¹⁴⁹ the code, adaptation of the code is not covered by the exception; i.e., adaptation of the code requires the authorization of the copyright holder, since adaptation is not required in order to derive information.¹⁵⁰

Additionally, three further conditions must be fulfilled cumulatively.¹⁵¹ First, the acts must be performed by the licensee or by another person having a right to use a copy of the program, or on their behalf by a person authorized to do so (Article 6(1)(a)). Second, the information necessary to achieve interoperability must not previously have been readily available to the persons who might otherwise legitimately perform the acts of decompilation (Article 6(1)(b)). As a third condition, the acts of decompilation must be confined to the parts of the original program which are necessary in order to achieve interoperability (Article 6(1)(c)).

3. A Compromise Called “Interoperability”

Because reverse engineering, including decompilation as by-product, implies potentially harm for the owners of the original program, the Directive¹⁵² in turn

(b) the information necessary to achieve interoperability has not previously been readily available to the persons referred to in subparagraph (a); and

(c) these acts are confined to the parts of the original program which are necessary to achieve interoperability.”

¹⁴⁸ CZARNOTA & HART, *supra* note 52, at 77.

¹⁴⁹ Only “transformation” of the form of code and not a “translation” into another language is subject to the exception of Article 6. *See id.*, at 77.

¹⁵⁰ *Id.*

¹⁵¹ Software Directive, *supra* note 10, art. 6(1), at 45, reads in full text:

“The authorization of the rightholder shall not be required where reproduction of the code and translation of its form within the meaning of Article 4 (a) and (b) are indispensable to obtain the information necessary to achieve the interoperability of an independently created computer program with other programs, provided that the following conditions are met:

(a) these acts are performed by the licensee or by another person having a right to use a copy of a program, or on their behalf by a person authorized to do;

(b) the information necessary to achieve interoperability has not previously been readily available to the persons referred to in subparagraph (a); and

(c) these acts are confined to the parts of the original program which are necessary to achieve interoperability.”

¹⁵² EU Software Directive, *supra* note 10, art. 6(2), at 45, reads as follows:

seeks to provide those owners with substantial protection by, first and foremost, narrowing the range of permissible purposes to interoperability.¹⁵³

3.1. *Interoperability—A General Definition*

Interoperability may be defined as “the logical and, where appropriate, physical interconnection...to permit all elements of software and hardware to work with other software and hardware and with users.”¹⁵⁴

For purpose of understanding the nature and significance of software interoperability, it may be helpful to distinguish two complementary levels of software:¹⁵⁵

- a) Operating system software consists of programs which operate low-level functions necessary for any use of a computer (particularly, the operating-system program), back the program creation (for example, compiler and test programs) and put common services (such as file management, file transfer, formatting) at disposal.
- b) Application (user) software, on the other hand, fulfills specific tasks like, for instance, word processing, spread sheet analysis, database creation and maintenance.

Operating system software (e.g., Microsoft’s Windows operating system or Sega Genesis machine) is the platform on which application software is designed to run (e.g., Lotus 1-2-3 or Sega’s games). Platforms and applications are not just complementary products; they are complementary parts of a system by virtue of

“The provisions of paragraph 1 shall not permit the information obtained through its application:

- (a) to be used for goals other than to achieve the interoperability of the independent created computer program;
- (b) to be given to others, except when necessary for the interoperability of the independently created computer program; or
- (c) to be used for the development, production or marketing of a computer program substantially similar in its expression, or for any other act which infringes copyright.”

¹⁵³ The idea behind Article 6 is that decompilation should not be permitted, but is unavoidable in certain strictly-defined situations. Recital 22 mentions that decompilation is allowed only under “limited circumstances”. It is interesting that in the Dutch language this was even translated as “rare (*zeldzame*) circumstances”.

¹⁵⁴ EU Software Directive, *supra* note 10, recital 10, at 43.

¹⁵⁵ HABERSTUMPF in LEHMANN, *supra* note 2, at 77.

their conformity to interfaces necessary for achieving interoperability.¹⁵⁶ In order to make the user software interact with the system software, the computer programmer must get access to the interface information of the system software, the so-called application programming interfaces (APIs).¹⁵⁷ It is the information underlying interfaces that provides the foundation of interoperability.¹⁵⁸

3.2. *How to Develop Software Interoperability—An Example*

In the 1980s IBM developed its first personal computer (IBM PC). The heart of the IBM PC is its Input-Output system (BIOS) which, in essence, operates the hardware (e.g., the printer port, monitor, keyboard and drives). Thus, in order to achieve interoperability with the IBM PC software developers had to emulate IBM PC-BIOS. However, IBM collaborated with the—in those days hardly known—software developer Microsoft to develop operating system software, MS-DOS, which was to be used as platform for the operation of user software. In other words, to exert control of the PC hardware user software had to interoperate with MS-DOS through the API.¹⁵⁹

As starting point, in the following an understanding of interoperability is provided in order to be able to comprehend the significance of interoperability in the computer industry and its consequences in the software decompilation context.

3.3. *Interoperability From An Economic Point of View*

¹⁵⁶ Samuelson & Scotchmer, *supra* note 5, at 1615-6.

¹⁵⁷ See, e.g., FELIX H. THOMANN, & GEORG RAUBER EDS., *SOFTWARESCHUTZ* [SOFTWARE PROTECTION] 10-1 (1998); John V. Howard, Jr., Jill Gushow, *Interoperability in the United States and Europe*, 24 *COLO. LAW.* 1275 (1995); Rotenberg, *supra* note 128, at 8.

¹⁵⁸ For illustrating purposes, the way how a word processing program notes in a text file that a certain word shall be printed in bold type, represents an example for such interface information. See JABUREK, *supra* note 120, at 21.

¹⁵⁹ For a detailed discussion see Vinje, *supra* note 32, at 251-3 (1992).

IBM sought to prevent software developers from having direct access to the functions embedded in its BIOS. By officially disclosing the necessary interface specifications of MS-DOS, software developers should have been able to create their own BIOS (i.e., tailor-made for their own PCs). However, the industry has begun to circumvent the use of Microsoft's interface specifications (APIs). Instead, software developers adjusted their user programs to the IBM-BIOS interface specifications. That is they made their user programs interoperable with the IBM-BIOS by passing by MS-DOS as platform where to achieve interoperability which was necessary to operate the PC hardware and as it was intended by IBM and Microsoft.

Software providers face various strategic options regarding interoperability. Some developers publish interfaces, some license them freely or against payment, and others maintain their APIs as closely held trade secrets. Firms may choose to keep their interfaces closed as a defensive measure, but also offensively as a means for capturing the market. What is the big deal about interface specifications? As the government in *United States v. Microsoft* made the point: “[T]o control the [interface specifications] is to control the industry.”¹⁶⁰ Software developers can make interface information available for two different purposes: horizontal and vertical interoperability.

3.3.1. Horizontal Interoperability

Horizontal interoperability means the availability of interface information that allows the reverse engineer to develop his own operating system while being compatible with already existing application software. Software providers disclose interface specifications of such quality and quantity as to enable subsequent software developers to analyze the interface information of the operating system necessary to develop independently their own operating system that is interoperable with other application programs. Therefore, horizontal interoperability leads to the development of independently created, competing operating systems which are compatible with other computer programs but not necessarily with the decompiled computer program itself. Applied to our aforementioned example that would mean that if IBM gives away the interface specifications of its IBM-BIOS required to accomplish horizontal interoperability, competing software developers (for example Apple) can develop their own BIOS on which all the user software, which was originally adjusted to the IBM-BIOS as *de facto* industry standard (such as Lotus 1-2-3 and WordPerfect), will run as well.¹⁶¹ With these facts in mind, new application software providers is given an incentive to write for the (dominant) MS-DOS platform, henceforth reinforcing its dominance.¹⁶²

¹⁶⁰ *United States v. Microsoft*, 65 F.Supp. 2d 1, 15 (D.D.C. 1999).

¹⁶¹ See Vinje, *supra* note 32, at 256.

¹⁶² Another example for horizontal interoperability occurred in the United States, namely in the case *Sony*, where horizontal access was accomplished through reverse engineering that led to the creation of an emulator that played Sony PlayStation games on a computer. See case cited *infra* note 455.

In terms of economic efficiency, developing a client PC operating system is a difficult, time-consuming, risky and expensive way to seek entry to the competitive market. Why do software engineers nonetheless undertake efforts to enter the horizontal market? With creating a new non-compatible operating system, thus with *a priori* no application able to run on it, software developers will encounter another barrier to enter the relevant market since users very unlikely will buy an operating system without a wide range of application programs already available, tested and used by other users.¹⁶³ However, horizontal interoperability with the result of making the new operating system interoperate with the existing body of application software, especially one that interoperates with the MS-DOS platform, can diminish those negative effects and, in addition, provide an incentive to undertake the costly and risky decompilation procedure.

3.3.2. Vertical Interoperability

In order to accomplish vertical interoperability software providers disclose interface information for their given operating system to application software providers to the extent to enable them to create new user programs independently.¹⁶⁴ The issue of vertical interoperability was recently discussed in the United States, in *Sega*,¹⁶⁵ where a competing manufacturer of game cartridges wanted to copy the protected standard embedded in the console as part of the reverse engineering process to ensure that his own-created games would work on that system. Vertical interoperability, in general, effects that in return those application programs will add value to the given platform.¹⁶⁶

In reality, endorsing the purpose of horizontal interoperability implies a threat for dominant providers of operating software that the computer market will open its

¹⁶³ See generally Rotenberg, *supra* note 128, at 11-3.

¹⁶⁴ Vertical Interoperability is illustrated in Vinje, *supra* note 32, at 256, as follows: By virtue of the same market dynamics that heaved the IBM-BIOS to the *de facto* industry standard for BIOS interfaces, MS-DOS has become the standard operating system program for the IBM-PC in those days. Digital Research reverse engineered MS-DOS with the result of marketing an operating system program, the so-called DR-DOS, in competition with MS-DOS. DR-DOS could operate the same functions as MS-DOS; i.e., DR-DOS interoperated with other software (such as Lotus 1-2-3) and hardware in the same way as MS-DOS. In other words, both operating system programs were interchangeable and thus competing with each other.

¹⁶⁵ See case discussion *supra* Part IV.2.3.

¹⁶⁶ Rotenberg, *supra* note 128, at 11-3.

doors to competing operating systems that were legally reverse engineered. Therefore, dominant software providers will be reluctant to uncover horizontal interoperability and rather induce vertical interoperability. Thus far, absent said otherwise in the Directive and given the Commission communication, the EU Software Directive appears to cover both horizontal and vertical interoperability.

3.4. *The Scope of Interoperability in a European Understanding*

Pursuant to Article 6(1) of the EU Software Directive, decompiling the code must be indispensable to obtain the interface information necessary to make software interoperable with the decompiled program.¹⁶⁷ One—probably the most important—purpose of Article 6 is, therefore, to act as a safety valve in the event that information enabling a second programmer to develop a program which can interoperate with existing programs is not available.¹⁶⁸ In spite of its significance at European level, there has been controversy during the passage of the Directive over the meaning of interoperability in two respects. While the decompilation exception codified in Article 6 is explicitly confined to interoperability, the Directive does not specifically permit decompilation for the purpose of research.¹⁶⁹ Whether this is intended or simply an omission is not clear.¹⁷⁰ As one commentator¹⁷¹ concludes that prohibiting research activities in the field of software development threatens to contravene Article 3(1)(n) of the EC Treaty, which lays down the Community objective to promote “research and technological development.”¹⁷² Furthermore,

¹⁶⁷ The final wording of Article 6 of the EU Software Directive expressly refers to the industry term “decompilation”—but not “reverse engineering”. Arguably, the Directive may not read too broad as to be intended to cover the process of reverse engineering as a whole.

¹⁶⁸ CZARNOTA & HART, *supra* note 52, at 76.

¹⁶⁹ Vinje, *supra* note 32, at 258 (stating that decompiling computer software for the purpose of research or other purposes is not permitted by Article 6).

¹⁷⁰ Guillou, *supra* note 109, at 533 (comparing the position under EU copyright law to copyright provisions of the U.S. Copyright Act which expressly permits reverse engineering for the purpose of research provided that the fair use test pursuant to section 107 is met); *but see* Verstryngne, *supra* note 78, at 10-4 (stating that decompilation for purposes which are limited to activities such as research was not intended by the Directive).

¹⁷¹ MARLY, *supra* note 111, at 328.

¹⁷² Article 3(1) of the Treaty Establishing the European Community, 2002 O.J. (L. 325) 33, at 41 [hereinafter EC Treaty], in the pertinent parts reads as follows:

“[T]he activities of the Community shall include, as provided in this Treaty and in accordance with the timetable set out therein:...

(n) the promotion of research and technological development[.]”

the Directive is silent as to whether or not the purpose of interoperability embraces not only decompilation to be allowed for creating attaching or “interfacing” products but also competitive replacement products for the after-market.¹⁷³

The Directive refers to interoperability broadly as functional interconnection and interaction as required to permit all elements of software and hardware to work with other software and hardware and with users in all the ways in which they are intended to function.¹⁷⁴ Somewhere else the Directive *expressis verbis* defines interoperability as “the ability to exchange information and mutually to use the information which has been exchanged.”¹⁷⁵ And so is the general objective of the decompilation exception rule expressed “to make it possible to connect all components of a computer system, including those of different manufacturers, so that they can work together.”¹⁷⁶ Overall, these insufficient definitions of the threshold criterion “interoperability” render the decompilation exception vague in terms of its scope and permissibility.

To linguistic inaccuracy was alluded for the first time when the United Kingdom proposed an amendment which, had it finally not been rejected in the Council Common Position,¹⁷⁷ would have permitted decompilation only to the extent necessary to create *connecting* programs, thereby excluding *competing* programs from the scope of the interoperability exception.¹⁷⁸ Following debates of

¹⁷³ Particularly, the lobbying groups representing customers and small and medium sized software companies pointed out the significance of a decompilation exception. Gaster, *supra* note 9, at 15.

¹⁷⁴ EU Software Directive, *supra* note 10, recital 10, at 43, reads in full text:

“Whereas the function of a computer program is to communicate and work together with other components of a computer system and with users and, for this purpose, a logical and, where appropriate, physical interconnection and interaction is required to permit all elements of software and hardware to work with other software and hardware and with users in all the ways in which they are intended to function[.]”

¹⁷⁵ *Id.*, recital 12, at 43, states in full:

“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[.]”

¹⁷⁶ *Id.*, recital 23, at 43, states in full text:

“Whereas an objective of this exception is to make it possible to connect all components of a computer system, including those of different manufacturers, so that they can work together[.]”

¹⁷⁷ See *Council Common Position* (EEC) No. 10652/1/90 of Dec. 14, 1990; also the Business Software Alliance (BSA) buttressed the standpoint of the Council reasoning that the British proposal could threaten the development of client PCs.

¹⁷⁸ MARLY, *supra* note 111, at 294 (indicating that under a fair dealing analysis, the permissibility of acts of decompilation turns on the motifs and purposes underlying the

both opposing arguments in the European Parliament,¹⁷⁹ the communication issued by the Commission to the European Parliament, contrary to the British recommendation, indicates that the term is to be construed as encompassing both compatible and competing computer programs.¹⁸⁰ Hence, a program developed through decompiling the target program may compete with that program provided that the decompilation is done for the purpose of achieving interoperability but not without facing harsh criticism.¹⁸¹

Another question that occurred in the European Communities is whether the purpose of interoperability may legalize newly developed programs to interoperate with the decompiled program only or with other programs in a way as to communicate and share data ("multi-vendor interoperability"). Hence, the independently created program would interoperate with the one which has been decompiled or with others in a system.¹⁸² Again the Directive is unclear. Arguably, 100 per cent compatibility in the sense that an independently created program performs all the functions offered by the decompiled program in exactly the same way, being virtually identical to that program is not demanded by the wording of Article 6 ("cloning").¹⁸³

Thus, to determine how broadly interoperability is to be understood, ultimately, lies with the implementing legislation and the courts of each Member State in years to

decompilation. Whereas acts of decompilation undertaken for non-commercial research purposes are permissible, the development of competing software based on the information of the decompiled program is not a purpose which will render the decompilation permissible under British copyright law).

¹⁷⁹ See, e.g., amendments proposed in the first reading period by Mr. Janssen Van Raay and Mr. García Amigo, 1990 EP DOC 142.529 (A3-173/17, 18, 21, 23).

¹⁸⁰ See *Communication from the Commission to the European Parliament of Jan. 18, 1991*, PARL. EUR. DOC. SEC (91) 87 FINAL-SYN 183, 5 (1991) (clarifying that "[d]ecompilation is permitted by Article 6 to the extent necessary to ensure the interoperability of an independently created computer program. Such a program may connect to the program subject to decompilation. Alternatively it may compete with the decompiled program and in such cases will not normally connect to it").

¹⁸¹ See, e.g., LEHMANN, *supra* note 2, at 22 (qualifying the purposes of cost-cutting or facilitating the development of competing programs as beyond of what is meant by the Directive under interoperability); Verstryngge, *supra* note 78, at 10-4 ("Article 6 does not however permit decompilation beyond what is necessary to achieve interoperability of the independently created program. It cannot therefore be used to create a program reproducing parts of a decompiled program having no relevance to the interoperability of the independently created program); CZARNOTA & HART, *supra* note 52, at 83.

¹⁸² WALTER, *supra* note 38, at 221.

¹⁸³ See CZARNOTA & HART, *supra* note 52, at 78.

come and cases to adjudicate,¹⁸⁴ whereby the Directive's objective to unify Member States' legislation is threatened to be undermined by diverse national case law. This paper advises Member States having a look at the international platform of copyright law to ensure a coherent interpretation within Europe, and thereby avoiding distortion of the Common Market.

3.5. *Is There Guidance By The Berne Convention, TRIPS and the WIPO Panel?*

Since all the Member States to the EU are at the same time signatories to the Berne Convention for the Protection of Literary and Artistic Works¹⁸⁵ and its progeny national legislators and judges may find clarifying guidance in those international copyright treaties. The Directive expressly mentions in its preamble "not [to] affect derogations provided for under national legislation in accordance with the Berne Convention on points not covered by this Directive."¹⁸⁶ Article 6(3) appears to have been introduced to advance efforts of interpretation of the EU Software Directive rather than to bring the Directive in line with the Berne Convention.¹⁸⁷

3.5.1. The Three-Step Test Under Article 9(2) of the Berne Convention

The need for a uniform system of copyright protection on the international level led to the formulation and adoption of the Berne Convention for the Protection of Literary and Artistic Works.¹⁸⁸ The aim of the Berne Convention, as indicated in its preamble, is "to protect, in as effective and uniform a manner as possible, the rights of authors in their literary and artistic works." Article 9(2) stipulates three distinct

¹⁸⁴ See Guillou, *supra* note 109, at 533.

¹⁸⁵ Berne Convention, *supra* note 33.

¹⁸⁶ EU Software Directive, *supra* note 10, recital 29, at 44.

¹⁸⁷ The ECIS argued that Article 6(3) was intended merely as a statement that the Directive is already fully consistent with the Berne Convention and not to confirm consistency with the Berne Convention. Had that been the case otherwise, the Directive's drafters would probably have included Article 6(3) in the recitals, not in an article on the narrow topic of decompilation. Haaf, *supra* note 23, at 424.

¹⁸⁸ The Berne Convention is the oldest international treaty in the field of copyright. The Berne Convention has been revised several times in order to improve the international system of copyright protection which the Convention provides. Changes have been effected in order to cope with the challenges of accelerating development of technologies in the field of utilization of authors' works, in order to recognize new rights and also to allow for appropriate revisions of established ones. The first major revision took place in Berlin in 1908, and this was followed by the revisions in Rome in 1928, in Brussels in 1948, in Stockholm in 1967 and in Paris in 1971.

conditions that must be complied with before an exception to the exclusive reproduction right can be justified under national law—the so-called “Three-Step Test”. According to the test exceptions are only allowed

- in certain special cases,
- provided that they do not conflict with a normal exploitation of the work, and
- provided that they do not unreasonably prejudice the legitimate interests of the authors.¹⁸⁹

The test sets limits to the national exceptions on the copyright holders’ reproduction rights. The three steps of the test are cumulative and, hence, a failure to comply with one of the steps results in the national limitation being disallowed. The importance of the Three-Step Test for the delicate balance between private and public interests in the field of copyright law is crucial. However, the broad language of each single prong blurs the actual meaning of the test as it would be required to adequately address the significant degree of guidance for its member nations.

3.5.2. The Three-Step Test Revisited By the WIPO Panel

Since its first mention in the Berne Convention of 1886, the tripartite test was, *inter alia*, also introduced in the TRIPS Agreement,¹⁹⁰ the WIPO Copyright Treaty¹⁹¹ and the EU Copyright Directive.¹⁹²

¹⁸⁹ Paragraphs (1) and (2) of Article 9 of the Berne Convention, *supra* note 33, say:

“Right of Reproduction: 1. Generally; 2. Possible exception; ...

(1) Authors of literary and artistic works protected by this Convention shall have the exclusive right of authorizing the reproduction of these works, in any manner or form.

(2) It shall be a matter for legislation in the countries of the Union to permit the reproduction of such works in certain special cases, provided that such reproduction does not conflict with a normal exploitation of the work and does not unreasonably prejudice the legitimate interests of the author.”

¹⁹⁰ Article 13 of the Agreement on Trade-Related Aspects of Intellectual Property Rights, [hereinafter TRIPS Agreement]:

“Limitations and Exceptions.

Members shall confine limitations or exceptions to exclusive rights to certain special cases which do not conflict with a normal exploitation of the work and do not unreasonably prejudice the legitimate interests of the right holder.”

Article 13 of the TRIPS agreement differs from Article 9(2) of the Berne Convention in two respects. First, under Article 13 the tripartite test applies to all exclusive rights, not only to reproduction. Second, the TRIPS test is restrictive in intent: i.e., Article 13 expressly restricts permissible limitations and exceptions to those which comply with its standards, while Article 9(2) of the Berne Convention also allows limitations based on other treaty provisions.

Despite the fact that the Three-Step Test was incorporated in each of those treaties, no considerable degree of agreement exists as to the actual meaning of the test since none of the treaties defines itself any of the significant terms used in the test, and the theoretical possibility of a dispute resolution by the International Court of Justice in this regard—as provided for in Article 33(1) Berne Convention—has never been used. However, in 2000 a WTO dispute resolution panel (WTO Panel)¹⁹³ dealt with the interpretation of the Three-Step Test as contained in Article 13 of the TRIPS Agreement and extensively analyzed each of the steps on the occasion of a dispute between the EU and the United States over an exception to the rightholders' copyright in U.S. copyright law.¹⁹⁴ As to the “certain special cases” requirement, the WTO Panel considered both quantitative and qualitative elements. Under the first, the unauthorized use in order to be justified has to be for a specific and designated purpose. The qualitative element requires the unauthorized use also to be “justified by some clear reasoning of public policy or some other exceptional circumstance.”¹⁹⁵ With regard to the second requirement of the Three-Step Test, the WTO Panel examined the meanings of the terms “normal” and “exploitation” as follows:

[N]ot every use of a work, which in principle is covered by the scope of exclusive rights and involves commercial gains, necessarily conflicts with a normal exploitation of that work. If this were the case, hardly any exception or limitation could pass the test of the second condition and Article 13 might be left devoid of meaning,

¹⁹¹ World Intellectual Property Organization Copyright Treaty art. 10, adopted Dec. 20, 1996, S. TREATY DOC. NO. 105-17, 36 I.L.M. 65, 71 (1997) [hereinafter WCT]:

Limitations and Exceptions.

“(1) Contracting Parties may, in their national legislation, provide for limitations of or exceptions to the rights granted to authors of literary and artistic works under this Treaty in certain special cases that do not conflict with a normal exploitation of the work and do not unreasonably prejudice the legitimate interests of the author.

(2) Contracting Parties shall, when applying the Berne Convention, confine any limitations of or exceptions to rights provided for therein to certain special cases that do not conflict with a normal exploitation of the work and do not unreasonably prejudice the legitimate interests of the author.”

¹⁹² See EU Software Directive, *supra* note 10, art. 6(3), at 45; see also discussion *infra* Part III.3.3.

¹⁹³ It has been the first and only decision rendered by a supra-national body, a WTO dispute resolution panel, established under Article 64 of the TRIPS Agreement, on the interpretation of the Three-Step Test in the context of Article 13 of the TRIPS Agreement. Unlike the Berne Convention, the TRIPS Agreement has teeth; i.e. the dispute-resolution mechanisms of the WTO stand ready to entertain allegations that the national laws of WTO countries are out of compliance with Article 13.

¹⁹⁴ See United States-Section 110(5) of the US Copyright Act, Report of the Panel WT/DS160/R, 15 June 2000 (hereinafter Homestyle Case).

¹⁹⁵ Homestyle Case, p. 32 et seq.

because normal exploitation would be equated with full use of exclusive rights....[A]n exception or limitation to an exclusive right in domestic legislation rises to the level of a conflict with a normal exploitation of the work..., if uses, that in principle are covered by that right but exempted under the exception or limitation, enter into economic competition with the ways that right holders normally extract economic value from that right to the work (i.e., the copyright) and thereby deprive them of significant or tangible commercial gains.¹⁹⁶

In this context, both actual and potential uses should be considered. Regarding the third step, the WTO Panel did not limit the notion of “legitimate interests” to actual or potential pecuniary interests or lawfulness but also to objectives that underlie the protection of exclusive rights. After all, the third step contains an important proportionality test in the way that the harm to the rightholders has to be reasonably related to the users’ benefits. In other words, the prejudice has to be proportionate.¹⁹⁷

Due to the fact that the Three-Step Test is formulated independently from technological advances, it may serve as a suitable instrument in international copyright law to examine the legitimacy of national copyright limitations within the digital environment. This leads to investigating the relationship of the tripartite test with Article 6 of the EU Software Directive.

3.5.3. The Three-Step Test under Article 6(3) of the EU Software Directive
Absent precedent case law in Europe on the questions of decompilation and reverse engineering of computer programs, Article 6(3) was introduced in the EU Software Directive. Article 6(3) stems from the Three-Step Test originating in Article 9(2) of the Berne Convention.¹⁹⁸ Taking Article 9(2) of the Berne Convention as a model, Article 6(3) of the Directive ensures that the decompilation exception shall not be “used in a manner which unreasonably affects the rightholder’s legitimate interests or conflicts with a normal exploitation of the computer program.”¹⁹⁹ It sets a test

¹⁹⁶ *Id.*, at 44 et seq.

¹⁹⁷ *Id.*, at 57 et seq.

¹⁹⁸ See *Commission Report*, *supra* note 36, at 13.

¹⁹⁹ EU Software Directive, *supra* note 10, art. 6(3), at 45, reads in full text:

“In accordance with the provisions of the Berne Convention for the protection of Literary and Artistic Works, the provisions of this Article may not be interpreted in such a way as to allow its application to be used in a manner which unreasonably prejudices the right holder's legitimate interests or conflicts with a normal exploitation of the computer program.”

against which national courts can judge whether decompilation has been acceptable or not in the circumstances of paragraphs (1) and (2).

Against the backdrop that, as for the time being, six EU Member States (that is Austria, Denmark, Finland, the Netherlands, Sweden and the United Kingdom) have been failed to implement Article 6(3), the Commission is concerned that the omission of any such explicit limitation provided for in Article 6(3) could result in unreasonable detriment to the software copyright holder, thereby re-opening the floodgate for diverse national case law by unbalancing the delicate compromise that was attempted to strike the decompilation exception of Article 6.²⁰⁰ However, the Commission's concern may be negligible. All EU Member States are also member nations of the Berne Convention. That means that each Member State must give effect to national legislation which complies with the Berne Convention. The source of the obligation to implement Article 9(2) into national copyright law is thus the Berne Convention itself, irrespective of Article 9(2)'s reiteration in Article 6(3) of the EU Software Directive. Therefore, the existence and significance of Article 6(3) is not at all clear. Rather its meaning has become subject to different interpretative approaches. As a result of literal interpretation, the purpose of Article 6(3) would be to safeguard that if interpretations of the preceding subparagraphs (1) and (2) led to outcomes contradicting with paragraph (3), then the standard as set forth in the latter, in the understanding of Article 9(2) of the Berne Convention, would be meant to prevail.²⁰¹ Assuming that, over time, national courts will develop diverse interpretations of the paragraphs (1) and (2), paragraph (3) is certainly added for extra guidance in interpretation.²⁰² Another purpose may be drawn from the facts that the direct applicability of the Berne Convention may differ from one Member State to the other, and that computer programs subject to decompilation may have their origin in an EU Member State but which, at the same time, is not a signatory to the Berne Convention as well—

²⁰⁰ See *Commission Report*, *supra* note 36, at 13.

²⁰¹ See Vinje, *supra* note 32, at 258; see also Dreier, *supra* note 54, at 325 (stating that Article 6(3) is superfluous and thus reducing the provision's meaning to serve as a security precaution against an inappropriate implementation of Article 6(3) by the Member States).

²⁰² CZARNOTA & HART, *supra* note 52, at 82 (arguing that paragraph (3) shall enable judges to assess whether the acts of decompilation in question would be acceptable under paragraphs (1) and (2)); MARLY, *supra* note 111, at 325 n.349 (citing Lehmann who argues that Article 6(3) is meant to serve as tool in the hands of judges when misuse in cases involving acts of decompilation is at issue).

thus the Three-Step Test of Article 9(2) of the Berne Convention would not be applicable automatically to such works.²⁰³ After all, the drafters of the EU Software Directive intended to ensure internationally a minimum level of protection for all computer programs being subject to decompilation under the Directive.²⁰⁴

Finally, applying the Berne minimum standard in interpreting the scope of interoperability would mean that decompilation would not permit unreasonable interference with the rightholder's exploitation of the computer program.²⁰⁵

Accordingly, the Directive would bar the use of information obtained by way of decompiling the target program which is subsequently reverse engineered for the purpose of creating a competing program. In other words, the scope of the decompilation exception would be limited to the purpose of vertical interoperability. Advocates of a broad understanding of the interoperability purpose in Article 6 define vertical interoperability as not only to imply the cloning of computer programs but also the multi-vendor interoperability.²⁰⁶ To their view, interoperability would not be restricted to any particular program. This view encounters harsh criticism by opponents, such as IBM, who opine that Article 6 then will enable third programmers to produce cheap imitation programs. They believe that such practice would clearly run counter to Article 9(2) of the Berne Convention.

In conclusion, the position taken by the Commission, whereby interoperability is defined to include the creation of competing programs (i.e. horizontal interoperability), and by the framers of the Berne Convention and the WCT (i.e. vertical interoperability in a narrow sense) seem to contradict each other – a puzzle each single Member State now has to deal with.²⁰⁷

²⁰³ WALTER, *supra* note 38, at 231.

²⁰⁴ Verstrynghe, *supra* note 78, at 10-4 (“We do not want to see the boundaries which we have drawn extended in a way which would cause the exception to fall outside the scope of Article 9.2 of the Berne Convention. If this were to be the case, then we would have achieved less protection for computer programs in Europe rather than more, and that was not our objective.”).

²⁰⁵ Spoor, *supra* note 39, at 1070.

²⁰⁶ CZARNOTA & HART, *supra* note 52, at 78.

²⁰⁷ See, e.g., MARLY, *supra* note 111, at 325 (outlining the different ways of interpretation of Article 6(3) of the Directive by different commentators and suggesting that given the obscure wording and its actual meaning what a Member State best could do is to implement the formulation of Article 9(2) of the Berne Convention into the Member State's jurisdiction—however, without giving a concluding statement).

3.6. *Decompilation Beyond the Need for Interoperability*

3.6.1. Maintenance of Computer Programs

Examples for software maintenance may be translation of computer programs written in an outdated programming language into a modern program language, or transcription of computer code from an amorphous form (“spaghetti code”) into a structured language, to regain the source code once lost or to port application programs onto new hardware. Furthermore, decompilation may be of importance for computer security. Through acts of decompiling application software, such as encryption software or authority checking systems, for security reasons, the users would be enabled to check whether their software is free of Trojan horses or “back doors”.²⁰⁸ The question is whether the Directive does leave room for acts of decompilation to be performed or as a means to track down interface information but likewise as a means for software maintenance.

Even though the final wording of Article 6 misses any reference to permissible decompiling computer programs for the purpose of software maintenance, it is the legislative documentation that shows evidence for intentionally not mentioning maintenance of computer software as legitimate purpose for decompiling computer software. The Commission devised the first draft reading “indispensable to achieve the creation, maintenance or functioning of an independently created interoperable program”²⁰⁹ as far too extensive and consequently the idea behind Article 6 was reduced to the sole purpose of accomplishing interoperability.²¹⁰ Since Article 6 now permits decompilation to generally “achieve interoperability”, users may argue that the broad meaning of Article 6 also covers acts done in order to maintain a computer program’s interoperability with other programs. However, unlike the proposal by the European Parliament,²¹¹ the Commission has not confirmed maintenance *per se* (i.e., maintenance for purposes other than interoperability) as an independent reason justifying decompilation. Especially, updating or improving

²⁰⁸ WALTER, *supra* note 38, at 212.

²⁰⁹ *Proposed Directive*, *supra* note 31, at 28.

²¹⁰ See *Council Common Position*, *supra* note 177, has replaced the passage in question as follows: “...indispensable to obtain the information necessary to achieve the interoperability of an independently created computer program with other programs.”

²¹¹ See Parliament Amendment, *supra* note 126, at 35; see also Amended Proposal, *supra* note 126, art. 5(a).

the target program is reserved to be done only by the copyright owner of the target program and not the licensee.²¹² Alternatively, the exception of Article 5(1) of the EU Software Directive may be apply to maintenance of computer programs, reading that “[reproduction and translation of computer code] shall not require authorization by the rightholder where they are necessary for the use of the computer program by the lawful acquirer in accordance with its intended purpose.”²¹³ Which acts are in accordance with the target program’s intended purpose? Likewise it is not clear, which, if any, acts of maintenance *per se* are covered by Article 5(1) provided that they are in accordance with the target program’s intended purpose.²¹⁴

For identifying the intention of the software copyright licensor the licensee may search the license agreement or rely on the purpose of the computer program licensed as generally ascribed to in computer software industry. After all, Article 5(1) obviously provides only an unsteady basis for *per se* maintenance of computer programs as permissible interoperability purpose. Still another legitimate question is whether error correction falls within the scope of software maintenance as computer users, arguably, assume that it is a computer program’s inherent purpose to run without faults.²¹⁵

3.6.2. Error Correction

A software user may cite Article 5(1) of the EU Software Directive as expressly permitting reproducing, translating, adapting, or otherwise altering a computer program to analyze errors by eliminating them in the program without authorization of the copyright holder provided that the error correction is necessary for the use of the program.²¹⁶ Furthermore, likewise without authorization by the

²¹² LEHMANN, *supra* note 2, at 18, 18 n.78.

²¹³ EU Software Directive, *supra* note 10, art. 5(1), at 44.

²¹⁴ Dreier, *supra* note 54, at 322, 324 (proposing that acts of maintenance shall be covered by Article 5(1) as long as they are in accordance with a program’s intended purpose but without clarifying which acts and how the conditions could or should be met).

²¹⁵ MARLY, *supra* note 111, at 329 (interpreting computer maintenance as, possibly, to include error correction within the original program writer’s intended purpose).

²¹⁶ EU Software Directive, *supra* note 10, art. 5(1), at 44, reads in full text:

“In the absence of specific contractual provisions, the acts referred to in Article 4 (a) and (b) shall not require authorization by the rightholder where they are necessary for the use of the computer program by the lawful acquirer in accordance with its intended purpose, including for error correction.”

rightholder, the user may permanently store and subsequently run the altered program (i.e., the program then without errors).²¹⁷ In determining so, Article 5(1) appears to render the purpose of interoperability obsolete when copying a computer program for error correction. All the more, the error correction provision is realized to be rather vague, especially as it gives neither a definition of an error correction nor the computer bug itself.

Error correction means all the acts appropriate to track down, locate and fix a bug.²¹⁸ A computer bug within the meaning of Article 5(1) might be defined as a software fault in a technical sense.²¹⁹ Regardless of an overall definition of a computer bug, while one user may identify an operation of a program as an error, another user may define the same operation as an intended feature.²²⁰

Some form of decompilation will often be indispensable to analyze and subsequently correct errors. Like in the case of software maintenance, the purpose of error correction may be asserted as long as the user can show that the program copy was “necessary“ to achieve interoperability.²²¹ Acts of error correction must be triggered by the existence of a concrete and acute computer bug. If, for example, research shows that the problem why compatibility cannot be accomplished does not lie in the reverse engineer’s own software, he would probably have no option but to try and discover through acts of decompilation what peculiar interaction between his software and the other software causes the crash.²²² Otherwise each

²¹⁷ HABERSTUMPF in LEHMANN, *supra* note 2, at 160 (stating that the process of analyzing errors may need fixing the program onto a paper or data medium which is also covered by the exception of Article 5(1)).

²¹⁸ CZARNOTA & HART, *supra* note 52, at 65.

²¹⁹ As most prominent example may be quoted the Y2K problem (the year 2000 problem also known as the millennium bug and the Y2K Bug) which was the result of a practice in computer program design that caused some date-related processing to operate incorrectly for dates and times on and after January 1, 2000 because they stored years with only two digits and thus the year 2000 would be represented by “00” and could also be interpreted by software as the year 1900.

²²⁰ See Kathleen Gilbert-Macmillan, *Intellectual Property Law for Reverse Engineering Computer Programs in the European Community*, 9 Santa Clara Computer & High Tech L.J. 247, 261.

²²¹ *Cf. id.* (stating that the decompilation restrictions of Article 6 do not apply to error correction under Article 5(1) rather the limitations of the decompilation provisions ought to be diminished provided that decompilation was “necessary” for the intended use of the program, including error correction).

²²² Johnson-Laird, *supra* note 7, at 889.

user could engage in decompilation by circumventing the need of the copyright owner's consent with reference to the potential necessity of error correction.²²³

The question for users, therefore, is whether one shall better comply with the standard set forth in Article 6 or Article 5(1). The problem is that the relationship between Article 5(1) and Article 6 is not at all clear. It follows from the provision of Article 5(1) that error correction may not exceed what is needed in order to fix a computer bug. In computer reality, users will face the problem of knowing beforehand exactly where the correction will need to be made. Thus, users may well be forced by nature to decompile larger parts than where "straight" decompilation takes place under Article 6.²²⁴ On the other hand, under the guise of error correction, software developer may be invited to "free-riding" decompilation. Because the legislators have dealt so perfunctorily with the subject of error correction, a free-rider problem may be at the wake. In addition to Article 5(1)'s vagueness, the Directive lacks also a provision forbidding any further use of the information obtained in the course of permissible error correction. Given the vague wording of Article 5(1) European counsels might advise users to abide with the decompilation restrictions of Article 6 when engaging in acts of error correction. However, in any event, error correction for the mere purpose of gaining an economic advantage is not deemed sufficient.²²⁵ As to software providers, European counsels may suggest for entering into maintenance agreements or offering new releases of their computer programs on a regular basis because only then they would entirely avoid the error correction problem in advance.²²⁶

Notably, Article 5(1) is construed as default rule as the text expressly provides for other contractual arrangements. Again no further indication is given whether error correction may be entirely forbidden by contractual provisions. This vagueness has already led to different outcomes in the laws of the Member States. For example, the Portuguese Copyright Statute altogether forbids contracting out of the provision. As for my understanding of the concept, such rigid approach is in line with the end of Article 5(1) which is to warrant a user's right to use, including

²²³ WALTER, *supra* note 38, at 200.

²²⁴ Spoor, *supra* note 39, at 1069.

²²⁵ WALTER, *supra* note 38, at 196-7.

²²⁶ JONGEN & MEIJBOOM, *supra* note 6, at 13; *see also* WALTER, *supra* note 38, at 187.

reproduce, a program according to its intended purpose, including error correction. Thus, a contractual provision prohibiting a lawful user from error correction under all conditions is highly questionable to comport with the Directive's intended objective. Another approach has been chosen by British case law that has carved out an implied right to correct errors.²²⁷ After all, it is indeed remarkable that the provision has not led to more debate in the European Parliament.

3.6.3. Research

Once more the EU Software Directive does not take an explicit stand on whether decompiling a computer program for the purpose of research is beyond the notion of interoperability or not. With the aim to shed light on the exact scope and permissibility of the interoperability requirement, the jurisprudence in the United Kingdom shall provide guidance given the Directive's lack regarding the research dilemma in the software decompilation context.

Before implementation of the EU Software Directive, the general rules on the U.K. doctrine of fair dealing, which is similar to the U.S. fair use concept, applied also to the purpose of research.²²⁸ As the EU Software Directive as well as its legislative history is silent with regard to research, the Directive called for changes in the copyright law having been in force in the United Kingdom at that time.²²⁹ Accordingly, the fair dealing rules were replaced by the exception of the Directive allowing decompilation solely to the extent necessary to achieve interoperability of

²²⁷ See *British Leyland Motor Corp. & Ors. v. Armstrong Patents Company Ltd. & Ors.* (1986) 2 W.L.R. 400 (Feb. 27, 1986) (U.K.), available at <http://www.bailii.org/uk/case/UKHL/1986/7.html> (last visited at Sept. 18, 2007).

²²⁸ § 29(1) of the Copyright, Designs and Patents Act 1988 (U.K.) [hereinafter 1988 Copyright Act] reads as follows:

“Fair dealing with a literary, dramatic, musical or artistic work for the purpose of research or private study does not infringe any copyright in the work or, in the case of a published edition, in the typographical arrangement.”

²²⁹ The main exception to a software owner's right to copy and adapt a computer program under the 1988 Copyright Act is that, subject to any contractual right or obligation, the “fair dealing” with a computer program for the purpose of research, private study, criticism or review or reporting current events is not infringement of copyright. The term “fair dealing” is not defined in the 1988 Copyright Act, although a likely test for fair dealing is whether it would be reasonable to expect that a person making a copy of a part or the whole of a work, should purchase the work instead. See JONGEN & MEIJBOOM, *supra* note 6, at 231; for a presentation of the British fair dealing doctrine see also MARLY, *supra* note 111, at 293.

computer programs.²³⁰ Outside the scope of software decompilation, however, the fair dealing defense may well apply to cases not involving computer programs. Thus, in excluding software from the scope of the traditional fair dealing provisions it appears that the United Kingdom has construed the terms of the Directive as barring the use of information obtained through decompilation for research.²³¹ Arguably, the structure of U.K. copyright law thereby has turned inconsistent and so judges may well incline to ignore the narrow scope of interoperability in the understanding of the European legislators.

At the end of the day the Directive's vagueness with regard to the scope of interoperability as the crucial condition for permitting decompilation of computer programs threatens to slow down the harmonization of all Member States' legislations in the field of software decompilation.

3.7. *Copyright Protection of Know-How as Unintended Consequence of Interoperability*

The need for a compromise prompted the European legislator to overly restrict the purpose for decompilation to interoperability. Article 1(2) of the EU Software Directive provides that "ideas and principles which underlie any element of a computer program, including those which underlie its interfaces, are not protected by copyright under this Directive."²³² In line with the mainstream concept pursued in the European Communities, originality is the decisive factor in determining the scope of copyright protection. Those aspects of interfaces will not be protected if they do not meet the originality threshold. However, Article 6 seems to ignore that deeply rooted copyright principle because according to Article 6, decompilation is prohibited for any purpose other than achieving interoperability—irrespective of whether copyrightable or non-protectible elements are at stake. Why should acts of decompilation—beyond the purpose of creating interoperable computer programs—constitute copyright infringement when used only to determine the

²³⁰ See Copyright (Computer Programs) Regulations 1992 (S.I. 1992, No. 3233), §§ 50A, 50B, 50C (U.K), which are operative from January 1, 1993.

²³¹ For a detailed discussion of implementing EU directives into U.K. copyright law see Guillou, *supra* note 109.

²³² EU Software Directive, *supra* note 10, art. 1, at 44.

underlying ideas, and not to copy protected expression? Then principally uncopyrightable know how contained in computer programs would be protected outside the scope of Article 6.²³³ Such narrow interpretation of Article 6 would result not only in a deviation from general copyright principles²³⁴ but also constitutes, as for the time being, a unique copyright approach in the European Communities.²³⁵

4. The Indispensability Criterion

Decompiling the computer program must be indispensable to obtain the information necessary to achieve the interoperability of an independently created computer program.²³⁶ The indispensability criterion of Article 6 of the EU Software Directive in terms of decompilation is interpreted in a restricted sense as to be of *ultima ratio* for software developers, who have no other way as to rely on the very interface information necessary for the creation of interoperable programs. In other words, Article 6 acts as a safety valve,²³⁷ i.e., only if the indispensable information is not available in any other non-infringing way, software developers shall be entitled to copy the code and transform it without authorization of the rightholder.

²³³ LEHMANN, *supra* note 2, at 23 n.90 (given the pressure purposefully exerted by interested cycles on the Commission, above all the software producers, the Directive embraces the protection of know how and trade secrets embedded in a software's interface specifications).

²³⁴ WALTER, *supra* note 38, at 217.

²³⁵ Technical protected privileges do not provide for such wide extent of protection. The Semiconductor Chip Protection Directive excludes in Article 5(3) reproduction for the purpose of analyzing and evaluating the concepts, processes, systems or techniques embodied in the topography or the topography itself from authorization by the rightholder. Moreover, Article 5(1)(b) refers solely to commercial exploitation and importation for that purpose of a topography or of a semiconductor product manufactured by using the topography (which is the exclusive right of the rightholder). *See* Council Directive 87/54/EEC of 16 December 1986 on the legal protection of topographies of semiconductor products, 1987 O.J. (L 24) 36 (Jan. 27, 1987). These provisions are in accordance with the purpose immanent in technical protected privileges, namely, to make the idea underlying the invention immediately accessible to the public in order to contribute to innovation.

²³⁶ EU Software Directive, *supra* note 10, art. 6(1), at 45.

²³⁷ CZARNOTA & HART, *supra* note 52, at 76.

In other words, a software programmer cannot opt for the process of decompilation in preference to other, non-infringing measures available.²³⁸ This view is also substantiated by the Explanatory Memorandum to the Amended Proposal where it is stated that

...as a last resort, a person having a right to use a copy of a program may commit acts of reproduction and translation of the machine-readable form of the code in which the copy has been supplied without the authorization of the rightholder, subject to certain limitations. ...There seems to be no justification for a policy which permits authors' rights to be infringed when circumstances do not demand it. The amended proposal therefore limits the application of the exception to circumstances where non-infringing means are not adequate.²³⁹

Article 6 indicates that there are different methods of how interoperability may be accomplished—some of them, apparently, are infringing while some of them are not. What does “indispensable” mean? However, the EU Software Directive is silent as to which non-infringing technical alternatives to decompilation are available and which (practical) role each of them plays in daily life of the software industry. Neither does the Directive regulate the chronology of events in the process of software creation, but relies instead on the ability of the defendant to justify his actions at any given moment.²⁴⁰

The requirement for “interoperability of the independently created computer program” provided in Article 6(2)(a) is to ensure that any decompilation of a target program does not occur before the independently created program exists. It does not matter whether decompilation of the target program occurs before the final coding of the independently created program into machine-readable form, or even at some earlier stage of the development cycle. But in any event the new program must already exist, at least, in the form of preparatory design material.²⁴¹ Therefore, if the creator of the new computer program has no preparatory design work completed on his own program before he begins decompiling the target program, it

²³⁸ *Id.* (distinguishing the *ultima ratio* approach of Article 6 from the fair dealing/fair use exceptions insofar as the latter provide that circumstances will justify the performance of otherwise restricted acts, regardless of whether other means exist to accomplish the same objective).

²³⁹ *Commission Preparatory Memorandum*, *supra* note 24, at 10-1.

²⁴⁰ CZARNOTA & HART, *supra* note 52, at 79.

²⁴¹ *See Commission Report*, *supra* note 36, at 14; *see also* CZARNOTA & HART, *supra* note 52, at 84 (requiring a “paper trail” to demonstrate the existence of the program before decompilation was embarked upon).

may be difficult for him to justify why decompilation was indispensable at that given stage in the development process, or to prove that the information was not readily available by other means, since he will not have himself defined the interoperability which his program will seek to achieve with those of other software programmers. In light of efficiency considerations, however, the second software engineer does not have to complete the coding into machine-readable form of all the interfaces of his independently created program before attempting to achieve interoperability.²⁴²

4.1. *Analyzing Preparatory Design Material*

As one alternative for finding routes to interoperability, reverse engineering may be accomplished by reviewing specifications, manuals and other technical material to the extent they are (publicly) available.²⁴³ The source code itself rarely is disclosed, only documentation containing information thereto. Such documentation may be included with the computer program or otherwise available from the software's manufacturer. Based on those documentation then independently computer programs may be developed.²⁴⁴ Here, the fact that reverse engineering is, by all accounts, a costly and time-consuming practice comes into play.²⁴⁵ Before a reverse engineer resorts to more difficult and more expensive practices of reverse engineering, such as decompilation, he probably has sought the information required to develop a compatible product from the documentation released by the original software developer.²⁴⁶

Despite of the advantages of analyzing preparatory design material, the truth is that documentation usually proves to be incomplete, inaccurate, and out-of-date when compared to the actual software itself.²⁴⁷ By its very nature and the manner of its production, the documentation is a statement of intent, of how the actual computer

²⁴² CZARNOTA & HART, *supra* note 52, at 79; WALTER, *supra* note 38, at 222.

²⁴³ JONGEN & MEIJBBOM, *supra* note 6, at 16.

²⁴⁴ Bayha, *supra* note 8, at 179-80.

²⁴⁵ CZARNOTA & HART, *supra* note 52, at 76.

²⁴⁶ Vinje, *supra* note 32, at 253 n.22 (IBM published the entire source code of its first PC-BIOS and manuals concerning further information of its PC-BIOS interfaces).

²⁴⁷ For example, a look at the "Readme" file, included in most software programs, will disclose all of the errors in and changes made to the documentation of the program after the manual was made. See Prestin, *supra* note 40, at 154.

program should be, rather than how it is.²⁴⁸ Therefore, computer bugs ordinarily are not mentioned, much less, described in manuals. The inaccuracy of documentation has two consequences. First, in order to achieve full interoperability also unexplained operations must be reconstructed. Second, if one is motivated to decompile a program because of an unexpected failure in its operation, and the intent is to understand why the program does not perform, rather than to accomplish full interoperability, then the software developer will be urged to avail himself of other efficient practices.²⁴⁹

Assuming that the analyzing of preparatory design material does in fact constitute a sufficient source for the software developer's purpose there is still the question of the legal status of analyzing preparatory design material under copyright law. Notably, in Article 6 the term "code" is used, whereas somewhere else in the Directive the term "computer program" appears. Albeit in computer science both terms are synonyms, this is not true in the context of the EU Software Directive. For the purpose of the EU Software Directive²⁵⁰ the term "computer program", in contrary to the term "code", includes preparatory design material.²⁵¹ Therefore, the exception rule of Article 6, by way of referring to "code" but not to "computer program", does not include the duplication and reconstruction of preparatory design material.²⁵² The same is true for other items normally considered to fall within the meaning of computer program, such as user manuals, computer output and computer databases. Those likewise remain to be protected or not according to each Member State's domestic copyright law.²⁵³ Notwithstanding the confusing terminology used in the Directive, or perhaps just because of lack of clarity, none

²⁴⁸ Johnson-Laird, *supra* note 7, at 846, 860 (characterizing the documentation as a "word picture of the [actual] program, not the program itself").

²⁴⁹ See Vinje, *supra* note 32, at 253.

²⁵⁰ Article 1(1) of the EU Software Directive, *supra* note 10, at 44, reads in full text:

"1. In accordance with the provisions of this Directive, Member States shall protect computer programs, by copyright, as literary works within the meaning of the Berne Convention for the Protection of Literary and Artistic Works. For the purposes of this Directive the term 'computer programs' shall include...preparatory design material."

²⁵¹ Preparatory design material must refer "to the development of a computer program provided that the nature of the preparatory work is such that a computer program can result from it at a later stage." See *id.*, recital 7, at 42.

²⁵² CZARNOTA & HART, *supra* note 52, at 77; see also WALTER, *supra* note 38, at 219.

²⁵³ BAINBRIDGE, *supra* note 4, 121 (making a copy of a flowchart used in the design of a computer program will infringe the copyright in the computer program under European copyright law, while under U.K. law the flowchart would be protected independently as an artistic work).

of the EU Member States has provided for a legal definition of computer program in their national copyright laws. After all, the assorted terminology as chosen by the drafters of the EU Software Directive indicates that the intent pursued in the Directive was not to restrict the reading of disclosed documentation in the same way as the process of decompiling computer software.,²⁵⁴ thereby regarding the analyzing of preparatory design material in broader terms as non-infringing means to access of interface information.

4.2. *Black-Box Analysis (Article 5(3) of the EU Software Directive)*

A computer program is input into a computer, and in operating a black-box analysis, the manner in which the computer functions and the output which is generated is observed while being run.²⁵⁵ The black-box analysis may take several forms. For example, a black-box analysis can be done by intermediately loading the program in the computer memory, running it, and then viewing the screen displays.²⁵⁶ Those ideas and functional concepts underlying word processing programs, spreadsheets, and video game displays are readily discernible without need for decompilation because the external expression of the object code is visible on the computer screen. However, those utilitarian elements of computer programs that run in the background, such as basic input-output system programs (BIOS) and operating systems, cannot be readily seen during the operation of the program. Thus, another method of the black-box analysis is running the program in an emulated environment, thereby observing the operation of the program through the use of another program known as a “debugger”.²⁵⁷ Through either process, without

²⁵⁴ Vinje, *supra* note 32, at 252 n.19, 253.

²⁵⁵ Meninsky, *supra* note 100, at 594 (by capturing the input and output of the computer program, a software engineer may be able to try to isolate any commonalities or patterns that emerge, particularly at start-up time; in a further step the software developer mimics the exact sequence observed in his new product).

²⁵⁶ Spoor, *supra* note 39, at (nailing down black-box testing as an attempt to find out from the outside how the computer program works in the inside).

²⁵⁷ See Prestin, *supra* note 98, at 154-5; *see also Sony*, 203 F.3d at 600-1.

A “debugger” is a computer program that, when run in conjunction with the application program, permits the computer software engineer to trace through the application program and observe how the program works. The “debugger” aids a software engineer to delve into the inner workings of a microprocessor as it executes an application program. *See Morrison*, *supra* note 130, at 312 n.110.

ever looking at the source or object code of the program,²⁵⁸ it may be possible to determine how the program is designed and what it accomplishes, thereby enabling a programmer to produce (i.e., reverse engineer) a program which operates similarly.²⁵⁹ From the technical perspective, research and diagnosis by means of a black-box analysis is principally based on mere reasonable guesses as a result of the observations by the software developer.²⁶⁰ From the legal perspective, it is noteworthy that, unlike acts of decompilation, a black-box analysis *ex lege* is not limited to ends of interoperability. The challenge for copyright lies with another, more technical, aspect of the black-box analysis. Regardless of the method of observation used, the computer program is copied each time the reverse engineer boots up the computer, and requires the computer, as an intermediate step in the black-box reverse engineering process, to copy the program into computer memory. As a rule, copying the computer program to be analyzed is subject to Article 4(a) of the EU Software Directive according to which the permanent or temporary reproduction of a computer program by any means and in any form, in part or in whole, whether caused by loading, displaying, running, transmission or storage of the computer program, requires the rightholder's authorization to do so.²⁶¹ Basically, the aforementioned black-box techniques do constitute acts enumerated in Article 4(a).²⁶²

Hence, the black-box analysis to be a legal means in terms of copyright law requires an exception to the general rule.²⁶³ The exception for black-box analyses is laid down in Article 5(3), where observation, study or testing the functioning of the

²⁵⁸ Whenever the process entails the transformation of object code into source code, the process is subject to Article 6 of the EU Software Directive. Vinje, *supra* note 32, at 253 n.24.

²⁵⁹ Bayha, *supra* note 8, at 179.

²⁶⁰ One German commentator characterizes Article 5(3) of the Directive as an "Experimentierklausel". See MARLY, *supra* note 111, at 271; see also Johnson-Laird, *supra* note 7, at 862 (describing a black box observation as the making of reasonable guesses based on the observations done by the reverse engineer).

²⁶¹ EU Software Directive, *supra* note 10, art. 4(a), at 44.

²⁶² For example, a test run requires loading and running of a computer program according to Article 4(a) and also "hex dump" (i.e. studying the hexadecimal object code) requires loading onto the main memory in order to be displayed on screen.

²⁶³ EU Software Directive art. 5(3), at 44, reads as follows:

"3. The person having a right to use a copy of a computer program shall be entitled, without the authorization of the rightholder, to observe, study or test the functioning of the program in order to determine the ideas and principles which underlie any element of the program if he does so while performing any of the acts of loading, displaying, running, transmitting or storing the program which he is entitled to do."

program without committing infringements of the author's exclusive rights is permitted.²⁶⁴ Without the exception the performance of a black-box-analysis, including intermediate copies, would be subject to the authorization of the rightholder. Another exception to the general rule is found in Article 5(1)²⁶⁵ which measures the legality in terms of the intended purpose of a computer program. The question is whether there should be drawn a legal distinction between how to use and how to analyze a computer code. To put it another way, is a black-box test is covered by the exception set forth in Article 5(1). The answer may be no when such analysis is understood to go beyond of what is regarded necessary for the use of the computer program in accordance with its intended purpose.²⁶⁶ The answer may be yes when Article 5(3) techniques are thought to be included in the use of the target program copy.²⁶⁷

In conclusion, for some software black-box testing may be insufficient and decompilation may become necessary²⁶⁸ in order to understand how the program receives its instructions and how it works.²⁶⁹ A software programmer's attempts to use decompilation to obtain information on sequence, structure and organization of the program will be measured as to whether non-infringing means under Article 5(3) could have revealed the same information.²⁷⁰ Under such circumstances, "indispensable" methods such as decompilation are thought to come in lawfully, namely under the shelter of Article 6. However, conversely, Article 5(3) can never

²⁶⁴ WALTER, *supra* note 38, at 216.

²⁶⁵ EU Software Directive art. 5(1), *supra* note 10, at 44, reads in full text:

"In the absence of specific contractual provisions, the acts referred to in Article 4 (a) and (b) shall not require authorization by the rightholder where they are necessary for the use of the computer program by the lawful acquirer *in accordance with its intended purpose*, including for error correction" (emphasis added).

²⁶⁶ WALTER, *supra* note 38, at 198-9.

²⁶⁷ ULLRICH, *supra* note 76, at 75 (affirming that approach by reasoning that otherwise it would be a tautology since the person who has the right to use a copy of the target program anyway is entitled to uncover uncopyrightable ideas and principles underlying the target program).

²⁶⁸ Black box analysis involves subjecting a program to certain input conditions, and monitoring its output in an attempt to better understand its function. Decompilation on the other hand is a much more intrusive act, involving the conversion of the machine-readable object code back to source code. See Marc A. Ehrlich, *Fair Use or Foul Play? The EC Directive on the Legal Protection of Computer Programs and Its Impact on Reverse Engineering*, 13 PACE L. REV. 1003, 1005 (1994).

²⁶⁹ Guillou, *supra* note 109, at 533.

²⁷⁰ CZARNOTA & HART, *supra* note 52, at 78.

be applied to permit acts of decompilation.²⁷¹ Essentially, the ambit of Article 5(3) is restricted to black-box analyses, whilst Article 6 solely applies to acts of decompilation. As a consequence of the distinction, Article 5(3) is not confined to the end of interoperability (“*Rückwärtsanalyse*”). Article 5(3) exempts analyses of ideas and principles²⁷² underlying “any element of the program” by observing, testing or studying the user surface (“*Oberflächenanalyse*”).²⁷³ The general purpose behind Article 5(3) is to grant minimum access to unprotectible ideas and principles underlying a computer program. The software supplier therefore may well provide for such access through techniques others than those enlisted in Article 5(3). However, Article 5(3) must not go beyond of what is allowed under Article 6 as the outermost limit.²⁷⁴

4.3 *The Burden of Proving Indispensability*

The indispensability requirement of Article 6 requires proof that means others than decompilation to achieve the same objective were not available or technically not sufficient for the software developer’s purposes. In practice, available alternatives to decompilation are likely to precede acts of decompilation because decompiling a computer program, by its nature, is a lengthy, costly and inefficient procedure.²⁷⁵ Notwithstanding the negative connotation of decompiling a computer program, Article 6 lacks any provision as to who shall bear the burden of proof and what are the rules when decompilation is not indispensable to access the unprotected information desired. In the absence of an explicit provision at European level it will remain with the Member States to regulate who shall bear the burden of proving “indispensability”.²⁷⁶

²⁷¹ Vinje, *supra* note 32, at 253 (Article 5(3) was introduced as compromise between those industry groups in favor of allowing decompiling computer software and those against it).

²⁷² Article 5(3) distinguishes between observing program functions which is permitted and studying of the program itself which is not permitted. However, since the Article follows the general copyright rule that ideas and principles are not copyrightable subject matters, the search for ideas and principles underlying the program itself is covered by the rule is no copyright infringement. MARLY, *supra* note 111, at 272.

²⁷³ ULLRICH, *supra* note 76, at 75-6 (aptly headlining the provision of Article 5(3) “program analysis” and that of Article 6 “interoperability analysis”).

²⁷⁴ *Id.*, at 76-7.

²⁷⁵ *Commission Preparatory Memorandum*, *supra* note 24, para. 3.14, at 8.

²⁷⁶ Vinje, *supra* note 32, at 257 n.53.

Assuming that Article 6 intends to put the burden of proof upon the defendant software engineer²⁷⁷ (i.e. the one who lawfully decompiles an original computer program) the defendant must show that previously other means have been reviewed with the result that they proved to be inadequate as to achieve the same end in a non-infringing way.²⁷⁸ Therefore, in any litigation process it will be necessary for the defendant programmer to demonstrate not only the extent to which his acts were in accordance with Article 6,²⁷⁹ but also the chain of events which led to his recourse to Article 6.²⁸⁰ In contrast to the word “necessary” used in the same sentence, “indispensable” suggests an absolute (objective) standard which will not be met if the decompilation was not necessary in the particular circumstances. For example, if the required interface information can be clearly identified by means of Article 5(3), or is documented in manuals or technical literature, then the defendant software developer must justify the decompilation as going beyond those identified interface areas.²⁸¹ Who, however, decides whether the interface information readily available in documentation provided by the original software supplier is sufficient, accurate and up-to-date as for the purpose of the legitimate software user?²⁸² Even if a software developer gets engaged in an Article 5(3) technique, he may encounter even another evidentiary problem, to wit, how to prove that the analytic black-box test did not amount in decompiling the target program.²⁸³

It is certainly true that it may be difficult in practice to demonstrate that acts of decompilation have taken place and if so that there was no other way available given the facts of a particular case. Judges will have to turn to and rely on expert witnesses in questions of evidence. Given the highly technical circumstances decompilation cases may entail there is an undeniable probability that courtrooms

²⁷⁷ Haaf, *supra* note 23, at 428 (indicating that the statute should place the burden of proof for showing that the use of information obtained through decompilation is exempted upon the party charged with the improper use).

²⁷⁸ CZARNOTA & HART, *supra* note 52, at 84 (suggesting to maintain records, similar to those kept as a consequence of “clean room” activities, if decompilation for interoperability purposes is to be used).

²⁷⁹ *Id.* at 84 (recommending to institute a controlled set of procedures in the decompilation process as an important precaution to show compliance with Article 6).

²⁸⁰ *Id.* at 77 (stating that Article 6 implies the presumption that the software developer who decompiles the original program has reviewed other means to achieve the same end and has found them to be inadequate).

²⁸¹ JONGEN & MEIJBOOM, *supra* note 6, at 16.

²⁸² Guillou, *supra* note 109, at 533.

²⁸³ Morrison, *supra* note 130, at 318.

become a battlefield for expert witnesses. From this perspective, the effectiveness of such an elaborate provision on decompilation as the European legislators chose to adopt in Article 6 of the EU Software Directive is not free from doubts.²⁸⁴

5. Legitimate Users

The Directive refers to two different kinds of legitimate users. Article 5(1) limits the right to use a computer program in accordance with its intended purpose to the “lawful acquirer,” whereas acts of decompiling a computer program under Article 6 may be performed by any “person having the right to use a copy of the computer program”. Neither of these formulations is defined in the Directive itself. The Commission, however, communicated its understanding of a “lawful acquirer” as a purchaser, licensee, renter or a person authorized to use the program on behalf of one of the aforementioned.²⁸⁵ Any purchaser, even if he just bought the computer software for the purpose of subsequent decompiling of the software, satisfies the lawfulness requirement.²⁸⁶ But as to a definition for a “person having the right to use a copy of a program” the Commission remained silent so that the scope of legitimate users under Article 6 may be interpreted in light of both terms in relation to each other. Presumably, a “lawful acquirer” is narrower in scope as it encompasses only purchasers, whereas a “person having the right to use a copy of a program” also includes employees of the lawful acquirer.²⁸⁷

However, decompilation by third parties, such as the employees of a lawful acquirer, requires the authorization from the legitimate acquirer of the computer software and must concern the development of a program for his purpose.²⁸⁸ This conception resembles on the “works made for hire” doctrine known under U.S.

²⁸⁴ See Dreier, *supra* note 54, at 325.

²⁸⁵ *Commission Report*, *supra* note 36, at 12; see also WALTER, *supra* note 38, at 186.

²⁸⁶ See also *Sega*, 977 F.2d at 1522-3.

²⁸⁷ Guillou, *supra* note 109, at 533; but see WALTER, *supra* note 45, at 197, 199 (arguing that both terms are to be read identical, whereby tracing back the inconsistent terms to the disputed legislative history of the Software Directive); JONGEN & MEIJBOOM, *supra* note 6, at 12 (opining that there are no reasons for an intentional differentiation between both terms by the Community legislator).

²⁸⁸ WALTER, *supra* note 38, at 222.

copyright law.²⁸⁹ In fact, the pertaining legislative history discloses that the notion of the U.S. doctrine was initially contemplated to be introduced in the EU Software Directive but at the end of the day the U.S. “works made for hire” doctrine did not make its way to the final wording of the Directive.²⁹⁰ Rejecting a “works made for hire” conception at European level, casts doubt whether the European legislators in fact intended the now subsisting two categories of legitimate users.

What is ultimately clear is that permissible decompilation may be performed only by the person who has a license or another right to use the software that is subject to subsequent decompilation, or by a third person who is authorized to do so on behalf of such legitimate user,²⁹¹ thereby excluding hackers or possessors of pirated copies from the scope of application.²⁹²

6. Previously Readily Availability

The necessary interface information must not have previously been readily available to the person wishing to make an interoperable program.²⁹³ The Directive does not provide for an exhaustive or illustrative list of how a computer program may be made previously readily available. It is thus far from clear what the European Community legislators really intended with the term “readily availability”.²⁹⁴

Let us assume a software developer has stipulated by contract that he will provide interface information to the user on first request. Later, in lieu of providing exhaustive manuals showing how the program performs the intended functions, the

²⁸⁹ 17 U.S.C. § 101 defines a work made for hire as “a work prepared by an employee within the scope of his or her employment.” The definition must be read in connection with § 201(b), which provides that “the employer or other person for whom the work was prepared is considered the author for purposes of this title, and, unless the parties have expressly agreed otherwise in a written instrument signed by them, owns all of the rights comprised in the copyright.”

²⁹⁰ *Parliament Amendments*, *supra* note 126, art. 2(4), at 79 (suggesting the insertion of the following article: “Where a computer program is created by an employee in the execution of his duties or following the instructions given by his employer, the employer shall be entitled to exercise all economic rights in the program so created, unless otherwise provided by contract.”).

²⁹¹ EU Software Directive, *supra* note 10, art. 6(1)(a), at 45.

²⁹² CZARNOTA & HART, *supra* note 52, at 79.

²⁹³ EU Software Directive, *supra* note 10, art. 6(1)(a), at 45.

²⁹⁴ Vinje, *supra* note 32, at 257 (concluding that the final interpretation will take place in European courts on a case-by-case basis).

software owner merely discloses a basic specification of the interface to the computer program. Under these circumstances, is the “readily availability” criterion met or is any attempt to decompile the program, even for the sake of achieving interoperability, considered an unlawful act? Does the Directive impose an obligation on software developers to make readily available *adequate* documentation on the technical specifications of the interfaces employed in their programs? Absent a contractual provision to provide interface information how much of an effort is required of a reverse engineer to obtain the information directly from the rightholder before concluding that it is not “readily available” and proceeding with decompilation?²⁹⁵ Under Dutch law, for example, the copyright holder in a computer program, on request of the lawful acquirer, can be required to disclose the information sought prior to decompiling the program.²⁹⁶

Furthermore, the question raises whether there should be a different standard for small independent software programmers as compared to large multinational computer companies with sophisticated technology²⁹⁷ and a greater financial flexibility.²⁹⁸ Decompilation is only one step in the process of reverse engineering. Decompilation is followed by independent programming using only unprotected elements. Even starting from unprotected information obtained through decompilation, the competitor must still independently code, debug, test, and market his product. Succinctly, decompilation by itself does not reduce the cost of marketing competing software anyway.²⁹⁹ Therefore, a different standard to be applied to competing companies with a financially weak background would not

²⁹⁵ Guillou, *supra* note 109, at 533.

²⁹⁶ *But see* JONGEN & MEIJBOOM, *supra* note 6, at 181 (denying such a contractual provision to be null and void under Article 9(1) of the EU Software Directive, which has not been implemented in the Dutch Copyright Act); *see also* ULLRICH, *supra* note 76, at 78 n.155 (arguing that Article 6 of the Directive establishes merely a program owner’s duty to omit behaviour which would prevent a third party from decompiling the owner’s original computer program; nonetheless, the program owner may decide by its own will to disclose the program’s interface specifications to a third party pursuant to Article 6(1)(b)).

²⁹⁷ Gilbert-Macmillan, *supra* note 220, at 259.

²⁹⁸ JONGEN & MEIJBOOM, *supra* note 6, at 181 (equating a too high licensing fee in return for the information or overly expensive translation costs (i.e., to translate the information from a foreign language) with not being readily available as in the meaning of Article 6(1)).

²⁹⁹ Dennis S. Karjala, *Copyright Protection of Computer Software, Reverse Engineering, and Professor Miller*, 19 U. DAYTON L. REV. 975, 1014 (1994) (indicating that reverse engineering is in fact difficult and more time-consuming than designing a functional equivalent from scratch).

solve their actual problem given the total expenditure accruing for reverse engineering software.

One organization that has consistently advocated the Directive's decompilation provision to the limit, the ECIS, has insisted that national statutory implementation of the provisions should require that such interface information is to be made "available in published manuals or other material provided *generally to all owners or licensees*."³⁰⁰ Under the rule proposed by the ECIS, the dissemination of such detailed technical information would be limited to qualified developers upon request only because (i) unsophisticated end users could be intimidated by overly technical documentation not required for their use of a program copy, (ii) unlimited distribution would entail unwarranted expenditures, and (iii) would cause a severe drain on the software vendor's resources since it would support unqualified users to exploit the program copy.³⁰¹

What does "previously" mean in the given context? For example, the reverse engineer receives information adequate for the purposes of interoperability at the same time that he receives a copy of the program.³⁰² May it be presumed that the information is readily available, even though not previously? The purpose behind this provision is the promotion of using open systems and standard interfaces.³⁰³

The Directive, moreover, does not comment on whether or not the interface information must be made freely available or if the software developer (that is the one supplying the interface information) can charge the licensee specifically for the information. One may especially think of the scenario when the rightholder seeks to sell the interface information of the original computer program before the reverse engineer has begun to decompile the original program.³⁰⁴ Irrespective of the obligation to grant access under the conditions of Article 6, there appears to be nothing on the face of the Directive to prevent a software producer from entering

³⁰⁰ Haaf, *supra* note 23, at 421-2 (citing the European Committee for Interoperable Systems, Implementation in UK Legislation of EC Software Directive: Issues of Concern 6 (Sept. 1991) (emphasis added)).

³⁰¹ *Id.* at 422 (concluding that a blanket distribution of such detailed technical information would not only be counterproductive and uneconomical, but also beyond the Directive's aims).

³⁰² CZARNOTA & HART, *supra* note 52, at 79.

³⁰³ WALTER, *supra* note 38, at 223.

³⁰⁴ Vinje, *supra* note 32, at 257.

into an agreement for the supply of such information³⁰⁵ and demanding payment each time such information is made available since it is clearly in the interests of both parties that dialogue should take place and that adequate and accurate information³⁰⁶ shall be provided from the source which is most likely to be in a position to make available sufficient, accurate and up-to-date information.³⁰⁷ In such case, the question of the reasonableness of the terms and conditions of payment may become a subject matter of evaluation in competition terms.³⁰⁸

On the other hand, if the licensee refuses to pay for the information, the original software supplier then can protect its rights in the program against reverse engineering, including decompilation, by non-disclosure of the interface information.³⁰⁹ The refusal to access due to financial circumstances will enable the original developer to circumvent the intention of the Directive to move towards standardization of interface implementations and open systems.³¹⁰ Furthermore, the exemption for reverse engineering in Article 6 would in fact eviscerate.³¹¹

In deference to such inconsistent interpretation, the dialogue does not have to be restricted to making the information available exclusively and specifically to the reverse engineer. The interface information may be published to a wider audience, or a general offer to provide such information may be made whereby the need for

³⁰⁵ On several occasions the European legislators refused to impose an obligation on the reverse engineer first to make an offer to buy the interface specification as prerequisite to be entitled to subsequently decompile the original program. *See, e.g., Verstryngne, supra* note 78, at 10-4 (without mentioning the legal role of license fees in the context of Article 6).

³⁰⁶ Gilbert-Macmillan, *supra* note 220, at 259-60 (stating that interfaces of computer programs are rarely neatly defined); *see also* WALTER, *supra* note 45, at 223.

³⁰⁷ MORITZ & TYBUSSECK, *supra* note 16, at 81 (demanding an obligation to pay a license fee for the right to decompile interface information; thereby referring to a decision of the German Federal Constitutional Court affirming such obligation in 1971).

³⁰⁸ CZARNOTA & HART, *supra* note 52, at 80.

³⁰⁹ ULLRICH, *supra* note 76, at 80 n.165 (refusing the approach that access to interface information shall be conditioned upon payment of a license fee since the willingness of the rightholder to disclose it only upon extra payment will undermine “readily availability” and thereby the purpose of limited, but reliable access to the interface information required); *see also* LEHMANN, *supra* note 2, at 23 (opining that “readily availability” excludes payment of a license fee, thereby referring to the French version of the pertinent wording reading “informations...facilement et rapidement accessibles”); *see also* HABERSTUMPF in LEHMANN, *supra* note 2, at 163 (stating that the payment of a license fee contradicts “readily availability” and offering the original program owner in return for the disclosure of his program’s interface specification sales increase and enhanced use of his own program caused by marketing the new compatible software products).

³¹⁰ Dreier, *supra* note 54, at 324; *see also* CZARNOTA & HART, *supra* note 52, at 80.

³¹¹ Gilbert-Macmillan, *supra* note 220, at 263 (arguing that the right of the developer to charge money for the interface information should be specifically and vehemently denied in the Directive and in the implementing legislation in order to protect the fundamental intent of Article 6 in favor of interoperable systems).

all other developers to explore the program of the first programmer in detail would be obviated. The second program developer would receive detailed information without the risks and costs involved in trying to derive that information by his own efforts and the original software developer would not risk having his program entirely copied in the course of reverse engineering by competitors. Then the net result of this incentive to engage in an open dialogue would be an accelerated move towards standardization and open systems.³¹²

7. Necessary Parts of the Original Program

According to Article 6(1)(c) the process of decompilation is “confined to the parts of the original program which are necessary to achieve interoperability.”³¹³ The purpose of decompilation is clarified in Article 6(1) as “to obtain information”, as opposed to expression or other elements of the program, and that information must be necessary to achieve interoperability. The key criterion is interoperability. The term “information” is not defined either in the text of the Articles or in the preamble. Absent a definition, the ordinary meaning of the term would suggest that symbols, numbers, addresses, or combinations of these in the form of protocols, including the sequence and timing of exchanges, will all be considered as “information”.³¹⁴ One may argue that a broad reading of “information” runs the risk of contradicting the freedom of ideas and principles as set forth in Article 1(2) saying that ideas and principles which underlie any element of a computer program

³¹² CZARNOTA & HART, *supra* note 52, at 80.

³¹³ EU Software Directive, *supra* note 10, art. 6(1)(c), at 45. The software developer who decompiles a computer program for a purpose other than achieving interoperability (e.g., simply to create a competing computer program) must not undergo acts of decompilation with view to uncover the decompiled program’s underlying ideas, principles or methods of operation. Subsequently implementing those elements in a competing computer program infringes the copyright of the original program. Article 6 does not apply then. But other techniques of reverse engineering covered by Article 5(3) of the EU Software Directive do not require an interoperability purpose. Therefore, analyzing ideas and principles under an Article 5(3) operation not in connection to accomplish interoperability does not constitute an infringement. Vinje, *supra* note 32, at 256-7 n.51.

³¹⁴ CZARNOTA & HART, *supra* note 52, at 78.

are not protected by copyright under the Software Directive.³¹⁵ Non-copyrightable ideas and principles *a priori* are not covered by the scope of Article 6(1).

Interoperability can only be established through interfaces information. Thus, “necessary” in the given context refers to the interface specifications as implemented in the target computer program to be decompiled in search for that information. The wording endorses decompilation of interface specification only to the extent required to enable the reverse engineer’s program to be compatible with the target program, and not decompilation of the entire program.³¹⁶ The underlying problem, however, is that the determination of the factual question concerning what and how much interface information is necessary for a reverse engineer to develop an interoperable program can ordinarily be assessed only *ex post* but scarcely *ex ante*.³¹⁷

Certainly, the search for information on interoperability will not be limited to those interface areas which the programmer of the original has designated and supported as interfaces, if these interfaces are not deemed sufficient in relation to the objective pursued by the reverse engineer.³¹⁸ The minimum amount and quality of information to be published is suggested to suffice for a reverse engineer to be able to create a fully compatible interfacing program part.³¹⁹

If the programmer of the original program does not evenhandedly provide the information or only to an insufficient quantity or quality, it is questionable, from a technical point of view, whether it is possible for a reverse engineer in advance to locate interface information exactly.³²⁰ Often the reverse engineer, first, needs to reveal the location of the interface information sought. In doing so, he may be urged to copy the entire program verbatim by loading object code into computer memory or by decompiling the object code into source code. By way of copying the entire program the intermediate copyist will inevitably also copy protectible

³¹⁵ See also Vinje, *supra* note 32, at 258 (stating a *de facto* protection of ideas by preventing access to those ideas, but in reality the provision complies with the industry practice).

³¹⁶ JONGEN & MEIJBOOM, *supra* note 6, at 16.

³¹⁷ Vinje, *supra* note 32, at 257.

³¹⁸ CZARNOTA & HART, *supra* note 52, at 80-1.

³¹⁹ *Id.* at 84 (identifying a complete list of interface protocols with information on critical timing and information location and sequences as sufficient for a reverse engineer to create his own interfacing sub-routine which will fully interoperate through the interface decompiled).

³²⁰ Guillou, *supra* note 109, at 533.

expression³²¹ of the original program's literal elements for the purpose of uncovering the unprotected interface specifications. The likelihood of copyright infringement occurs regardless of whether the amount of protectible expression is small.³²² Such protected (interface) information obtained in the course of decompilation may in fact be "legally misused".³²³ So long as the reverse engineer confines his acts of reproduction and transformation to those parts necessary to achieve interoperability, he, nevertheless, is permitted pursuant to Article 6(1)(c) to copy the entire original program, containing unprotected as well as protected elements, without infringing the rightholder's exclusive rights.³²⁴

In any event, as soon as in the course of search for the interface information those parts of the original program that are necessary to achieve interoperability are uncovered further acts of reproduction and transformation within the meaning of Article 6(1) must be confined to those parts as soon as they have been carved out.³²⁵ Another way to concede national courts more flexibility in applying a standard appropriate to the industry practice would be to interpret "necessary" in a meaning of "*reasonably* necessary".³²⁶ Otherwise, in cases where it would be discovered at a later stage of the decompilation process that a particular, lawfully-intended decompilation was not necessary to accomplish interoperability, national courts would be forced to render judgments in contradiction to the furtherance of an environment in which to promote the establishment of open systems.³²⁷

³²¹ EU Software Directive, *supra* note 10, recital 13, at 43: "Whereas, for the avoidance of doubt, it has to be made clear that only the expression of a computer program is protected and that ideas and principles which underlie any element of a program, including those which underlie its interfaces, are not protected by copyright under this Directive[.]"

³²² Lewis, *supra* note 91, at 567 (pointing out that the technical characteristics of software makes it exceedingly difficult for a reverse engineer to study an original software work without creating a copy of the work, most likely in its entirety, thereby infringing at least some "nugget" of protected expression).

³²³ Cf. WALTER, *supra* note 38, at 224.

³²⁴ Although only expression of a computer program is protected, allowing expression to be "legally misused" in the context of Article 6(1)(c) appears incoherent with this general principle of copyright law.

³²⁵ WALTER, *supra* note 38, at 224 n.423 (suggesting to clarify the scope of interoperability by including the word "insofar" in the first sentence of Article 6(1) as follows: "The authorization of the rightholder shall not be required where and *insofar* reproduction of the code and translation of its form within the meaning of Article 4(a) and (b) are indispensable to obtain the information necessary to achieve the interoperability[.]" (emphasis added)).

³²⁶ Gilbert-Macmillan, *supra* note 220, at 264.

³²⁷ Another aspect of the necessity requirement in the context of the essential facility doctrine is dealt with in Part VII.2.1.

8. The Restricted Use of Decompiled Information

Whereas paragraph 1 of Article 6 lays down the conditions for decompiling the original computer program itself, paragraph 2 deals with the conditions in relation to producing a computer program based on the results of the decompilation.³²⁸ Even when interface information has been lawfully obtained, the Directive provides for three restrictions on the use of the information obtained through acts done in accordance with Article 6(1). The restrictions appear to be understood as complementary provision to paragraph 1.³²⁹ The information may not be obtained for goals other than to achieve the interoperability of the independently created computer program (Article 6(2)(a)), and, furthermore, may not be disclosed to others, except when necessary for the interoperability of their own independently created computer programs (Article 6(2)(b)).³³⁰ In other words, the re-use of the information for purposes other than interoperability is prohibited; i.e., the information decompiled may not be used for creating the computer program itself.³³¹ Thus, Article 6(2) prohibits the second stage of the overall process of reverse engineering of computer software.³³²

It is worth noting that Article 6 of the Directive does not extend the further use of an original interface to the insertion thereof into a new computer program.³³³ The abolition of free reverse engineering as such is also expressed in Article 6(2)(c) wherein the re-use of the information obtained through decompilation is not permitted for the development, production or marketing of a computer program substantially similar in its expression, or for any other act which infringes copyright.

8.1. *Substantial Similarity of Expression*

³²⁸ CZARNOTA & HART, *supra* note 52, at 81.

³²⁹ Vinje, *supra* note 32, at 258.

³³⁰ WALTER, *supra* note 38, at 225 et seq. (the information obtained through lawful decompilation is not permitted “to be given to others, except when necessary for the interoperability of the independently created computer program”).

³³¹ Vinje, *supra* note 32, at 256-7 n.51.

³³² JONGEN & MEIJBOOM, *supra* note 6, at 16.

³³³ For distinguishing decompilation from reverse engineering *see supra* Part I. and Part II.2.

In Article 6(1) the Directive refers to the process of reverse engineering itself by restricting decompilation to be done for the purpose of interoperability of an “independently created computer program” only. In addition, Article 6(2)(c)³³⁴ excludes the information obtained through decompilation to be used “for the development, production or marketing of a computer program substantially similar in its expression, or for any other act which infringe copyright.” Other acts infringing the author’s copyrights in his program are, for instance, making a translation of³³⁵ or transmitting the target program without authorization or making derivative works which are not covered by the concept of development or marketing a program substantially similar in its expression.³³⁶ The rest of the wording is yet far from being unequivocal.

The fraction supporting an EU Software Directive which should outlaw software decompiling for the purpose of creating competing programs intervened in favor for a provision prohibiting the creation of a program which was in substance similar to the target program.³³⁷ A broadly drafted Article 6(2)(c) would embrace functional similarity as well as similarity of the program’s expression. With computer programs, functional similarity of the decompiled and the new program is, by its nature, accompanied by the process of decompilation because the new program usually performs the same functions as the decompiled program; e.g., interoperating with other programs or hardware in the same technical way. That means that functional similarity and thus even insubstantial similarity would prevent a software engineer every time from creating new interoperable programs

³³⁴ EU Software Directive, *supra* note 10, art. 6(2)(c), at 45, reads as in full text:

“2. The provisions of paragraph 1 shall not permit the information obtained through its application:

(c) to be used for the development, production or marketing of a computer program substantially similar in its expression, or for any other act which infringes copyright.”

³³⁵ *But see* Haaf, *supra* note 23, at 428 (suggesting the term should be defined to incorporate “similarity in structure, sequence or organization along the lines of the definition used in the *Whelan* decision, thus enabling the translation of a program from one programming language to another” under copyright protection).

In *Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.*, 797 F.2d 1222 (3rd Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987) [hereinafter *Whelan*], the Third Circuit defined a program’s idea broadly as its purpose or function. The court found substantial similarity existed between the formats and structures, file structures, screen outputs, and file sub-routines. *Id.* at 1234.

³³⁶ CZARNOTA & HART, *supra* note 52, at 81.

³³⁷ 1990 EP Doc. 142.529 (A3 – 173/21, 23) of July 5, 1990.

on the basis of the decompiled information. Ultimately, the final version of the Directive did not comply with that understanding. Rather, the Directive refers to the traditional copyright principle according to which substantial similarity of *expression* and not function is determinative for infringement of the target program's author's copyright.³³⁸

How much is "substantial"? Under European copyright law *de minimis* similarity does not constitute an infringement,³³⁹ e.g., the identical adoption of letters "IBM" into the interoperable code is not a copyright infringement in Europe.³⁴⁰ To determine whether the similarity is *de minimis*, the courts will assess the "substantiality" by a qualitative threshold and not a quantitative.³⁴¹ In other words, the fact how prominent a role a copyrighted element plays in the defendant's work will be decisive rather than the amount copied.

A commentator raised the noteworthy question whether copied bugs are always evidence of substantial similarity resulting in copyright infringement. His answer was no. Even if an allegedly infringing program includes the same error of logic as the original, copying that bug in the reverse engineered program does not necessarily conclude slavishly (i.e. substantial) copying of the original. Reverse engineering would be misunderstood to guard against error copying since the reverse engineer is not capable to know the bug, let alone, how to correct it.³⁴²

Does the "substantial similarity" affect only those parts of the program in which information from the decompiled work was used, or rather the program in its entirety? The *telos* of the Directive as a whole is confined to information obtained through acts of decompilation. Therefore, the *de minimis* standard of Article 6(2)(c)

³³⁸ Vinje, *supra* note 32, at 258.

³³⁹ The legal maxime of *de minimis non curat lex*—the law does not concern itself with trifles—applies. *De minimis* copying is copying that is so trivial and insignificant that no liability can result.

³⁴⁰ In order to achieve interoperability with the EGA (Enhanced Graphics Adapter), the PC-BIOS had to consist of the 3 letters "IBM". Different software developers chose different ways of achieving interoperability; for example, AST Research adopted in its BIOS code the following words: "IBM is mentioned for compatibility only", Quandram Co. inserted in its BIOS code "Note: Some code expects 'IBM' here!". Microsoft Word Version 2.0 wished to interoperate with a PC and therefore screened the IBM's EGA BIOS in search for the letters "IBM". Otherwise Microsoft Word would have concluded that the PC searched is simply equipped with a CGA meaning a limited black and white resolution. Vinje, *supra* note 32, at 259.

³⁴¹ *Id.*, at 259 n.70.

³⁴² Johnson-Laird, *supra* note 7, at 900.

shall apply in the decompilation context.³⁴³ The similarities of the interface information must be, first, insubstantial and, second, the consequence of the merger doctrine³⁴⁴ and not of copying protectible expression, or merely co-incidental similarity.³⁴⁵ Complementarily, the basic, to say default copyright test, if and to the extent applied in the European Communities, shall apply to the remaining parts of the reverse engineered program. The question is whether there actually is or should be established a uniform European standard other than the one referred to in Article 6(2)(c).

The Directive does not elaborate a European standard upon what would constitute substantial similarity of expression. As a result of the obscure wording of the Directive, the drafters of the Directive apparently took into account unharmonious protection among the Member States since it is likely that Member States will formulate and develop different substantial similarity standards. Unlike in Europe, the U.S. courts have already ruled on the issue of what standards shall be applied in determining whether two programs are in fact substantially similar.³⁴⁶ Thus, the EU legislator could prevent unharmonious consequences on the substantial similarity standard to be adopted in Europe by inspiration found in U.S. case law.³⁴⁷

³⁴³ *But see* ULLRICH, *supra* note 76, at 79 n.160 (applying the provision of Article 6(2)(c) generally to the entire computer program at issue).

³⁴⁴ *See infra* Part III.9.4.

³⁴⁵ CZARNOTA & HART, *supra* note 52, at 82.

³⁴⁶ The U.S. courts have been less than unanimous in introducing a conceptual framework to determine the proper scope of protection of computer code; i.e. in distinguishing between the idea and expression of literal and non-literal elements. In *Nichols v. Universal Pictures Corp.*, 45 F.2d 119 (2d Cir. 1930), Judge Learned Hand came up with the “abstractions” test to determine how much and what constitutes substantial similarity. Upon any work, there is a point in the series of abstractions where there is only the most general statement left which is no longer protected by copyright law. Otherwise the work could prevent the use of ideas, to which, apart from their expression, the property of the owner is never extended. *See* 45 F.2d 119, 121. In *Computer Associates v. Altai, Inc.*, 982 F.2d 693 (2d Cir. 1992), the Second Circuit adopted another inquiry based on the Learned Hand “abstractions” test. The *Altai* inquiry involves three stages of analysis: abstraction, filtration, comparison. First, one must abstract the program into various layers of generality. Here, in a manner similar to reverse engineering, a court should dissect the copyrighted program’s structure and isolate each level of abstractions in it. Second, the merger doctrine can be used to filter out those elements of the program dictated by efficiency or by factors external to the program itself—e.g., mechanical specifications, compatibility requirements—or those taken from the public domain. When abstraction and filtration are accomplished, the third step involves comparing the remaining elements with the corresponding elements of the defendant’s work. While the courts may use these tests in conjunction with the principles of merger, scenes a faire, and utility, these tests can be redundant and difficult to apply.

³⁴⁷ However, given the highly fact-specific case law and the failure of any court to formulate a “bright-line” approach in the United States, the EU legislators as well as Member States will face difficulties in deducing a general substantial similarity test from the U.S. copyright law model.

8.2. *The “Clean Room” Procedure—A Safeguard?*

Decompilation as a step in a “clean room” process was not debated at European level.³⁴⁸ The clean room procedure aims at developing a program compatible with or functionally identical to another program, the “target program”, in order to enable the creation of a competing program. Basically, the efforts of reverse engineering are split between two isolated groups. The first group of engineers decompiles the target program to understand its underlying ideas. Subsequently, a third party hands the interface specifications detailed in a written journal or log (so-called “Survey Sheets”) over to a second, independent group of engineers. Without ever having seen the target program, the latter then uses those ideas in the development of an end product.³⁴⁹ In so doing, the new computer program shall be supposed to be independently created, meaning without allowing any suspicion of having copied expression from the target program.³⁵⁰ Why are clean room procedures nevertheless a contentious practice in the software industry?

On the one hand, clean room procedures are cumbersome, highly documented and typically costly processes. Albeit no direct copying of a competitor’s computer code takes place, nevertheless, there may be need for decompilation when decompilation is necessary to uncover the functional specifications for the compatible competing program. Like the intermediate copyist, the first group of engineers cannot filter out the interface information underlying a copyrighted program without copying it.³⁵¹ Subject to the process of decompilation is potentially unlawful because it involves the making of an intermediate copy of the

³⁴⁸ In the U.S. case *Sega v. Accolade*, the Ninth Circuit discussed the “clean room” process in a reverse engineering context. See *Sega*, 977 F.2d at 1526.

³⁴⁹ See *DSC Communications Corp. v. DGI Technologies, Inc.*, 898 F. Supp. 1183, 1189 n.3 (N.D. Tex. 1995) (explaining the use of a “clean room” in the reverse engineering process); for further literature see also Vinje, *supra* note 32, at 259 n.64.

³⁵⁰ For a definition and a detailed discussion of the clean room strategy in the case *IBM v. Fujitsu* see MORITZ & TYBUSSECK, *supra* note 16, at 77-81.

³⁵¹ See generally Lewis, *supra* note 91, at 573-4 (as a result of the fact that the second group has no direct access to the expression embodied in the program examined by the first group the end product necessarily will be different from the original program to an extent sufficient to avoid infringement; thus employment of clean room processes might inhibit the ability of the copyright holder to seek an injunction and can also affect the amount of damages).

code which may include either the target program, or the decompiled code, or some variation of both.³⁵²

On the other hand, decompilation in a "clean room" may qualify for the exception under Article 6 of the EU Software Directive. The defendant may assert the missing access of the "clean" engineers to the expression of the target program in order to defy the infringement claim because consequently the second program must be assumed to be created independently and similarities in expression are merely fortuitous.³⁵³ That means that the defendant is hard-pressed to prove something negative; i.e., the defendant must show evidence that he has not copied expression into his own substantial or virtually identical computer program.³⁵⁴ The Ninth Circuit made clear that a clean room procedure does not provide the defendant engineer or software company with a *prima facie* defense. In *Sega*, the Ninth Circuit stated that a clean room process is regarded as neither a non-infringing³⁵⁵ nor an efficient alternative to reverse engineering computer software.³⁵⁶ Although a clean room procedure by its nature proves absence of access, U.S. courts will hold the new program created in a clean room process as copyright infringement, if all or part of both codes are strikingly similar unless the alleged reverse engineer can prove that his work was not contaminated by access to any of the protected elements in the original program.³⁵⁷ Assumably, the same level of burden of proof will challenge defendants in courts throughout Europe. After all, also a "clean" engineer is still left with the question of how much identical or similar expression will be too much. Unlike its appearance, therefore, a clean room procedure does not by itself imply legality.

³⁵² See *id*; see also Bayha, *supra* note 8, at 180-1 (despite the infringing character of reverse engineering at the stage of decompiling the target code during the clean room process, ideally, the lack of both access to the target program or the decompiled code and the resulting lack of substantial similarity with respect to the new competing program renders the latter to be a non-infringing final work).

³⁵³ See Vinje, *supra* note 32, at 259.

³⁵⁴ See MORITZ & TYBUSSECK, *supra* note 16, at 78 (doubting whether a defendant can ever be in the position to refute successfully the suspicion of copying expression since the defendant, at least in theory, could have had contact with engineers outside the clean room—concluding that a clean room procedure probably does not provide for a sufficient protection from asserted copyright infringement claims).

³⁵⁵ Under U.S. copyright law a finding of copyright infringement requires proof of access to the original computer program and substantial similarity between the new program and the decompiled.

³⁵⁶ See *Sega*, 977 F.2d at 1519.

³⁵⁷ See generally HOWARD B. ABRAMS, THE LAW OF COPYRIGHT § 14.01 (1993).

Another drawback for clean room procedures with view to the reverse engineered end-product results from the nature of computer software and its methods of operation. In order to achieve interoperability of interface information, software developers under certain circumstances (as it will be seen in the following Part) inevitably may be forced to use copyrighted elements of expression of the target program.³⁵⁸

8.3. *The Merger Doctrine under European Copyright Law*

With the so-called “merger doctrine” another approach has been adopted resembling the U.S. copyright legal regime.³⁵⁹ The provider of the standard potentially holds a monopoly position by excluding competing software companies from access to its (functional) standard interface specifications which they would need to develop interoperable products. The Directive aims at obstructing such monopoly pattern, first, by means of the decompilation exception set forth in Article 6. Second, with view across the Atlantic, the U.S. merger doctrine has proven as a solution for Europe, at the same time, with the effect to bring the Member States’ legislation in line. Under the U.S. model of the merger doctrine, when external constraints dictate expression, idea and expression have merged, and thus the expression should not be protected.³⁶⁰ With computer programs, in

³⁵⁸ Vinje, *supra* note 32, at 259.

³⁵⁹ See *Commission Preparatory Memorandum*, *supra* note 24, at 8:

“3.13. If similarities in the code which implements the ideas, rules or principles occur as between inter-operative programs, due to the inevitability of certain forms of expression, where the constraints of the interface are such that in the circumstances no different implementation is possible, then no copyright infringement will normally occur, because in these circumstances it is generally said that idea and expression have merged.”

Two years later, In a debate in the European Parliament on April 16, 1991, M. Bangemann, then the Vice-president of the Commission, reiterated the position of the Commission as to recognize the merger doctrine in European copyright law as follows: “The ‘merger doctrine’, according to which there is no copyright protection where an idea and its expression cannot be separated...is a permanent feature of copyright law [and] we do not need to mention it in the directive.”

³⁶⁰ In *Baker v. Selden*, 101 U.S. 99 (1880), as the seminal case establishing the merger doctrine under U.S. copyright law, the U.S. Supreme Court held that to the extent that a work is functional, it may be copied, as may those expressive elements of the work that must necessarily be used as incident to expression of the underlying ideas, functional concepts, or facts. In other words, when the underlying ideas can effectively be expressed in only one way, so to say the idea and expression have merged); see also *Sega*, 977 F.2d at 1524 (citing *Computer Assocs. Int’l, Inc. v. Altai, Inc.*, 982 F.2d 693 (2d Cir. 1992) and the CONTU Report).

generating an interface in the independently created program which is functionally compatible with the interface the reverse engineer has decompiled in the original program, it may happen that the programmer is forced, given the increasing trend towards standardized interfaces,³⁶¹ to copy protected expression as well. When there technically no other options exist, the reverse engineer may re-use the information gained by way of decompilation to produce expression which is identical to that of the decompiled program. Then reverse engineering and decompilation should be condoned as to allow reverse engineers to discover the unprotectible elements and subsequently use these elements in the same way as they are expressed in the original program in their own created computer programs. Certain features of interface implementations may be dictated by function and therefore are not copyrightable. The case might be different where the code implementing the interface showed sufficient originality. If there is copyrightable expression in the original program's interface specifications, but the reverse engineer cannot find an alternative expression, then the engineer will be constrained to use the same expression because technically he lacks an alternative choice to accomplish interoperability.³⁶² In such circumstances, when the same functionality cannot be achieved otherwise—i.e., by way of different expression—it may well be that there is in fact an expression in the first interface which cannot be separated from the principles or methods of operations on which it is based, because the underlying idea and its expression have merged into the one and only possible solution. By virtue of the merger doctrine, if portions of (copyrighted) interface code deemed absolutely necessary in order to achieve interoperability were taken, there would be found no infringement provided that the rest of the new program shows no substantial similarities to the code of the decompiled program, even though those elements of code are not dictated by function in a strict sense.³⁶³

However, in a recent decision the court held that a 37-bytes code was, in its entirety, copyrightable expression, disregarding arguments of merger and programming efficiency, because the court found that the program could have been written in a number of different ways. See *Lexmark Int'l Inc. v. Static Control Components, Inc.*, 387 F.3d 522, 533 (6th Cir. 2004).

³⁶¹ HABERSTUMPF in LEHMANN, *supra* note 2, at 163 (identifying the standardization of interfaces as reason for the need to re-use the identical or at least almost identical interface specification of the original target program).

³⁶² CZARNOTA & HART, *supra* note 52, at 82; see also Vinje, *supra* note 32, at 259 n.70.

³⁶³ Dreier, *supra* note 54, at 325.

With computer software, and especially in the case of operating system software, external constraints often dictate conformity with industry standards of format and expression.³⁶⁴ For example, in our aforementioned IBM example, the IBM-PC (and MS-DOS) has been established as *de facto* standard. In the years following 1981 the IBM-PC has become the standard for software developers, hardware designers as well as computer users. Due to IBM's strong foothold in the personal computer market, software developers and hardware designers relied on the IBM-PC interface specifications in creating their products; i.e., by rendering their products interoperable with IBM-PCs they wanted to cover the relevant market. Consequently, computer users preferred IBM-PCs because of the array of by-products available for IBM-PCs.³⁶⁵

In 1991, the EU Software Directive has declared the objective of promoting international standardization.³⁶⁶ In order to realize the goal of international standardization the Community and the Member States have to face two points worth of considering. First, as a result of standardization, smaller computer companies must abide by the standard to secure their economic survival in the computer market.³⁶⁷ In so doing, re-using both protected and non-protected elements of the original program's interface specification because of mere commercial reasons—and not due to the lack of technical alternatives—does not suffice as a motive to have the merger doctrine applied.³⁶⁸ Second, Article 6 appears to establish a mere technical test in order for a software engineer to be eligible to decompile a computer program.³⁶⁹ To what extent may efficiency considerations be permitted to condone decompiling computer software? In spite of

³⁶⁴ Cf. Meninsky, *supra* note 100, at 596 (pointing out the necessity to distinguish what a program does and how it does it because “[s]oftware to achieve a particular result may be written, or expressed, in many different ways, but to do so, it may contain many elements that are dictated by function, efficiency, standards, or by compatibility requirements”).

³⁶⁵ Vinje, *supra* note 32, at 251-2.

³⁶⁶ Commission Preparatory Memorandum, *supra* note 24, at 7 (recognizing a trend towards standardization of computer products); *see also* EU Software Directive, *supra* note 10, recital 9, at 43 (“Whereas the Community is fully committed to the promotion of international standardization[.]”).

³⁶⁷ For example, in 1983, after the IBM-PC had been established as *de facto* standard, DEC marketed its own, not IBM-PC compatible PC, the Rainbow 100, with the consequence that DEC finally had to yield the demand for IBM compatibility.

³⁶⁸ *See* CZARNOTA & HART, *supra* note 52, at 82.

³⁶⁹ *See, e.g.,* Verstrynge, *supra* note 78, at 10-4 (“...it was technically indispensable for [the software programmer] to reverse engineer...”).

the rigid technical prerequisite for decompilation, the requirement of indispensability³⁷⁰ will likely be interpreted falsely if it leads to disregard efficiency consideration. The legislative history of Article 6 exposes that an interpretation encouraging economic inefficiency would hamper competition of European software programmers vis-à-vis U.S. software developers.³⁷¹

8.4. *Re-Distribution of Information*

In contrast to subparagraph (a) of Article 6(2) which focuses on the achievement of interoperability, subparagraph (b) refers to the state of interoperability; i.e., the meaning goes beyond re-distribution of information obtained via decompilation for the sole purpose of creating software.³⁷²

Article 6(2) allows a software developer to make the copyrighted information obtained, to say interface information in the form of copyrighted codes, available to others only when necessary for the interoperability of his own independently created computer program.³⁷³ Paragraph 2 is to be understood in light of paragraph 1. By referring only to Articles 4(a) and (b) only reproduction and translation are made subject to Article 5(1) since the right of distribution is enlisted in Article 4(c).³⁷⁴

Publishing or trafficking in copyrighted information concerning existing original computer programs may harm the interests of the copyright owner. If, as a result of decompilation, the new computer program is interoperable with the target program, it probably has implemented the interface information of the target program. In this way, the interface information can leak away to third parties who wish their

³⁷⁰ See *supra* Part III.4.

³⁷¹ Vinje, *supra* note 32, at 257; see also ULLRICH, *supra* note 76, at 78 n.156 (citing Haaf who deprives software programmers of their right to decompile and reverse engineer software under efficiency considerations as counterproductive premise in light of Europe's move towards international standardization).

³⁷² WALTER, *supra* note 38, at 228.

³⁷³ With reference to our example from Part II.3.2., the reverse engineer, who has decompiled interface information of the IBM-BIOS for the purpose of independently creating his own PC-BIOS which is interoperable with the IBM-BIOS, may disclose that interface information in his manuals. The customers shall be enabled to run other application programs on the PC-BIOS of the second software developer as they would on the IBM-BIOS in order to be fully interoperable. See Vinje, *supra* note 32, at 258.

³⁷⁴ See WALTER, *supra* note 38, at 226 (criticizing the failure of a systematic approach in terms of a rightholder's protection of his exclusive rights).

software to be interoperable with the second program, and consequently with the original program. Hence, only the re-distribution of information necessary for a concrete, independently created program or for purpose of achieving the interoperability of such second computer program may be permitted.³⁷⁵ It would be overreaching to impose conditions on the reverse engineer but allow third persons future access to the information which he has then made public.³⁷⁶ In terms of financial consequences, the provision will likely entail disadvantages for small and medium software producers.³⁷⁷

Although Article 6 of the EU Software Directive has struck a reasonable balance between the interests of software owners on the one hand, and the need for compatibility on the other, in other respects it may well form too narrow a system. While the Directive permits error correction and interoperability purposes, it does not leave room for decompilation for other reasons which, *prima facie*, might at times also be justified. One might think of educational purposes, maintenance of software which is no longer supported by the rightholder, and the wish to prove through decompilation that certain third party software infringes the rights in one's own software. In conclusion, although not dealing with all of the emerging questions, the Directive is certainly a good starting-point to bring the law of the European Communities into line with technological development in this field.

IV. Reverse Engineering under U.S. Copyright Act of 1976: Let the Battle on the Video Game Market Begin!

1. The "Limited Use" Exception

1.1. Section 117(a) of the Copyright Act

³⁷⁵ Cf. Parliament Amendments, *supra* note 126, art. 5a(2)(c), at 81: "... the information retrieved may not be communicated to third parties except in so far as this is necessary for the operation of the second program." See also WALTER, *supra* note 38, at 228.

³⁷⁶ CZARNOTA & HART, *supra* note 52, at 81.

³⁷⁷ MARLY, *supra* note 111, at 320.

The limited use exception incorporated in section 117(a) of the Copyright Act allows the owner of a computer program copy to authorize or make a second copy or adaptation of the program “as an essential step in the utilization of the computer program” or “for archival purposes only” and only so long as the possession of the computer program is rightful.³⁷⁸ Therefore, making backup copies of the copyrighted computer program purchased does not constitute copyright infringement. But does the statute also cover intermediate copying for reverse engineering purposes? In essence, the given question is the U.S. pendant to the contentious scope of the Article 5(1) exception to cases involving software decompilation. Whereas in Europe the dispute is about what is the intended purpose of computer programs, in the United States it is all about whether decompilation constitutes an essential step in the utilization of computer programs. The answers to the question have been different with the U.S. courts and led to a Circuit split between the Fifth and the Ninth Circuit as to whether section 117(a) shelters the intermediate copyist from liability.³⁷⁹

1.2. *Vault Corp. v. Quaid Software Ltd.*

In *Vault Corp. v. Quaid Software Ltd.*,³⁸⁰ the Fifth Circuit held section 117 of the Copyright Act to include the intermediate copying used in the decompilation process as “an essential step in the utilization of the computer program.”³⁸¹ Plaintiff Vault Corp. (“Vault”) produces computer diskettes under the registered trademark PROLOK which are designed to prevent the unauthorized duplication of programs placed on them by computer software companies, Vault’s customers. The

³⁷⁸ The full text of 17 U.S.C. § 117(a) (1998) reads as follows:

“Making of additional copy or adaptation by owner of copy.

Notwithstanding the provisions of section 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, or

(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.”

(Amended Oct. 28, 1998, Pub.L. 105-304, Title III, § 302(1), 112 Stat. 2887).

³⁷⁹ See generally Lewis, *supra* note 91, at 586-9.

³⁸⁰ *Vault Corp. v. Quaid Software Ltd.*, 847 F.2d 255 (5th Cir. 1988) [hereinafter Vault].

³⁸¹ *Id.* at 261.

PROLOK floppy diskettes serve as a medium upon which computer companies place their software programs. The protective device placed on a PROLOK diskette by Vault ensures by way of scanning certain information data stored on the PROLOK diskette that cannot be altered or erased (similar to a “fingerprint”) that the computer operates the program recorded on a PROLOK diskette by one of Vault’s customers only after the computer has verified that the original PROLOK diskette, as identified by the “fingerprint”, is in the computer’s disk drive. Vault distributed every PROLOK package with a shrink-wrap license. The defendant Quaid Software Ltd. (“Quaid”), a software manufacturer, decompiled plaintiff’s PROLOK program by copying it into the memory of its computer and analyzed the manner in which the program operated. Quaid’s product of the reverse engineering was a diskette called “CopyWrite”, containing a feature called “RAMKEY” which unlocks the PROLOK protective device and facilitates the creation of a fully functional copy of a program placed on a PROLOK diskette. The process is performed simply by copying the contents of the PROLOK diskette onto the CopyWrite diskette which can then be used to run the software program without the original PROLOK diskette in a computer disk drive. Vault brought an action seeking to enjoin Quaid from selling RAMKEY asserting, among other things, that Quaid had infringed Vault’s exclusive right of reproduction by copying Vault’s program into its own computer’s memory for the purpose of developing a program, RAMKEY, designed to defeat the function of Vault’s program.³⁸²

1.3. *The Decision*

The Fifth Circuit based its judgment on the CONTU Report³⁸³ “as an expression of legislative intent.”³⁸⁴ In a discussion at great length, the court concluded that a

³⁸² *Id.* at 256-8.

³⁸³ In 1974 Congress established the National Commission on New Technological Uses of Copyrighted Works (the “CONTU”) to perform research and make recommendations concerning copyright protection for computer programs. In 1978, the CONTU issued its final report (the “CONTU Report”) including a proposal for the Computer Software Copyright Act. *See* FINAL REPORT OF THE NATIONAL COMMISSION ON NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS (July 31, 1978).

³⁸⁴ *See Vault*, 847 F.2d at 260-1 (regarding the legislative history of the 1980 Computer Software Copyright Act, the Committee Report is not very enlightening but merely mentions that

computer program to be run, first, must be necessarily copied into a computer's memory, and thus the copying for the purpose of "utilization of the computer program in conjunction with a machine"³⁸⁵ is permitted pursuant to section 117.³⁸⁶ However, the exception under section 117 to be applicable requires that the copy made "is used in no other manner", which was the target of Vault's contention. It is a fact that Quaid copied Vault's program into its computer's memory for the "express purpose of devising a means of defeating its protective function."³⁸⁷ Nonetheless, the Fifth Circuit rejected Vault's interpretation of the second requirement set forth in section 117(a)(1) as to permitting copying of a computer program only for the purpose of using the program for its intended purpose. The court reasoned that neither the express language nor congressional intent contribute to such statutory interpretation.³⁸⁸ Thus, on this count the Fifth Circuit, on appeal, affirmed the district court's judgment as to hold Quaid not to having infringed Vault's exclusive reproduction right under § 106(1) of the Copyright Act.³⁸⁹

1.4. *A Critical Analysis*

Similar to the respective wording of Article 5(1) of the EU Software Directive, section 117(a)(1) permits making a copy for the purpose of using it in its intended manner and "in no other manner". The missing legislative history on the point may be negligible with view to the given semantic approach as to evidence the legislator's intent. Applied to the facts at hand, it would have led the Fifth Circuit to the conclusion that Vault certainly did not intend to license its software for the purpose of enabling competitors to develop their own unlocking program.

A further guidance for interpretation can be found in international copyright treaties signed by the United States, such as the legal development of the Three-Step Test under the Berne Convention.³⁹⁰ Yet, the limited precedent value of the

the Act embodies the recommendations of the CONTU Report, and thus courts have been prompted "to rely on the CONTU Report as an expression of legislative intent").

³⁸⁵ 17. U.S.C. § 117(1) (1998).

³⁸⁶ CONTU Report at 31.

³⁸⁷ *Vault*, 847 F.2d at 257, 261.

³⁸⁸ *Id.* at 261.

³⁸⁹ *Id.* at 261, 270.

³⁹⁰ For an in-depth discussion *see supra* Part III.3.3.

WTO Panel's decision should be borne in mind as it binds only the parties to the legal proceedings. Neither other member nations nor domestic courts are bound by the decision. Nonetheless, the WTO Panel decision can provide valuable guidance not only to national legislatures enacting legislation to comply with the Three-Step Test but also to those interpreting existing legislation, especially in circumstances where a contentious piece of national legislation is at issue. However, none of the U.S. courts ever reviewed the Three-Step Test in the context of the limited use exception of section 117(1)(a) as to delimit the scope of the provision with view to intermediate copying.

In *Vault*, the Fifth Circuit did not at all address the distinction between a licensee to and an owner of a copy of a computer program. Although Quaid was a mere licensee of Vault's computer software, the court accorded the benefit resulting from section 117(a) to the defendant, disregarding that, according to the express wording of section 117(a)(1), only the "owner" of a copy of the computer program and not the licensees as well is subject to the exception.³⁹¹ In 1993, the Ninth Circuit in *MAI Systems Corp. v. Peak Computer, Inc.* spelled out that licensees to computer software do not qualify as "owners" of software as to be eligible for protection under section 117(a)(1).³⁹² In contrast to the EU Software Directive, U.S. courts have thus explicitly established the subcategories of "owner" and "licensee" under the umbrella term "lawful acquirer".

Given these ambiguities though software copyright owners have recently become vigilant and went to protect their copyrights by way of drafting restrictive software license agreements with the effect that U.S. software licensors have become creative in restricting the contractual language of their software license agreements so as to preclude finally any possibility of an affirmative defense under section 117(a)(1) invoked on part of the software vendee. In so doing, software license agreements provide that the licensee owns neither the copyrighted software nor the media on which it is contained.³⁹³

³⁹¹ Cf. Bayha, *supra* note 8, at 187.

³⁹² See *MAI Systems Corp. v. Peak Computer, Inc.*, 991 F.2d 511, 518-9 (9th Cir. 1993) [hereinafter MAI Systems].

³⁹³ Bayha, *supra* note 8, 187.

1.5. *The Response of the Ninth Circuit*

Four years after *Vault*, that time the Ninth Circuit, in *Sega Enterprises Ltd. v. Accolade, Inc.*, encountered the question whether section 117(a) of the Copyright Act protects only the use of the computer program as intended by the copyright owner. In substance the court left the question unanswered³⁹⁴ by simply stating that “[s]ection 117 does not purport to protect a user who disassembles object code, converts it from assembly into source code, and makes printouts and photocopies of the refined source code version.”³⁹⁵ In its reasoning the Ninth Circuit stated that such conduct would go beyond the intent of drafters of the CONTU Report.³⁹⁶ However, three years after *Sega*, the Second Circuit decided *Aymes v. Bonelli*.³⁹⁷ This time the court did go to the substance of section 117(a)(1). The defendant bought plaintiff’s software and, subsequently, adapted it in a way necessary for continuing use of the software in operating defendant’s business.³⁹⁸ Holding the defendant not liable for copyright infringement on grounds of the exception to the copyright owner’s exclusive rights laid down in section 117(a)(1) of the Copyright Act, the Second Circuit essentially based its decision on the following paragraph of the CONTU Report:

Because of a lack of complete standardization among programming languages and hardware in the computer industry, one who rightfully acquires a copy of a program frequently *cannot use it without adapting it to that limited extent which will allow its use in the possessor's computer*. The copyright law, which grants to

³⁹⁴ See *Sega Enterprises Ltd. v. Accolade, Inc.*, 977 F.2d 1510 (9th Cir. 1992) [hereinafter *Sega*].

³⁹⁵ *Id.* at 1520 n.6.

³⁹⁶ CONTU Report at 13 (stating that given the fact that operating a computer program entails loading the program into the computer’s memory a rightful owner of a program copy shall be entitled to use it freely without being subjected to copyright liability).

³⁹⁷ *Aymes v. Bonelli*, 47 F.3d 23 (2nd Cir. 1995) [hereinafter *Aymes*].

³⁹⁸ That the Second Circuit in *Aymes* did not expressly refer to acts of decompilation, albeit the facts show adaptations and reproductions of a computer program took place, may be due to the (rare) fact that in the instant case the decompiling did not result in a reversed engineered new software that was sought to be sold. The most common product of reverse engineering is elements of decompiled or disassembled information which will be recoded and integrated into the new independently created program for distribution purposes.

Although the terminology of “reverse engineering” and “decompilation” is used—probably mistakenly—interchangeably, in fact decompiling needs to be distinguished from reverse engineering, in that reverse engineering goes beyond decompilation. Reverse engineering is a two-step process: first the computer program is reverse analyzed (i.e., decompiled) and in a second, the forward programming, step rebuild on the basis of the result of the decompilation.

copyright proprietors the exclusive right to prepare translations, transformations, and adaptations of their work, should no more prevent such use than it should prevent rightful possessors from loading programs into their computers. Thus, a right to make those changes *necessary to enable the use for which it was both sold and purchased* should be provided.³⁹⁹

The CONTU Report inherently reached the following conclusions. First, the core contribution to the discussion whether section 117(a)(1) of the Copyright Act embraces the making of intermediate copies is that the *Aymes* court held plaintiff's adaptation right not to be infringed because the adaptations were to be done to enable the use of the computer program as intended by the plaintiff-software vendor and defendant-software vendee. Second, in determining the intent of the parties the *Aymes* court recognized that "transactions involving computer programs are entered into with full awareness that users will modify their copies to suit their own needs."⁴⁰⁰ The conversion of a program from one language in another, for instance, by means of decompilation or disassembly, to keep it abreast from year to year and to maintain viability of the original software in case of computer upgrades to accommodate successive generations of operating software systems must be taken into account by software vendors. But distribution of adapted programs in an unauthorized manner will be beyond the scope of section 117(a)(1).⁴⁰¹ After all, had *Vault* been decided in light of the *Aymes* rule the software vendees' intent to decompile the computer program for the purpose of subsequent distribution of the reverse engineered product could be argued by Vault to be beyond the scope of section 117(a)(1) as clarified in *Aymes*. Likewise, Quaid's behavior would not have been exempted under the EU Software Directive since re-distribution of the new program created on the basis of the information decompiled must be solely motivated by the purpose to accomplish interoperability with the new program. Pure economic purposes would not suffice for applying Article 6(2)(b).

1.6. Conclusion

³⁹⁹ CONTU Report at 32 (emphases added).

⁴⁰⁰ *Aymes*, 47 F.3d at 26.

⁴⁰¹ *Id.*

Under the *Vault* ruling, a competitor was permitted to create a means of copying another's copyrighted computer program free of charge, even though it was against the software copyright owner's intended use. In respect of the competitive impact therein, competitors were enabled to undermine the marketability of Vault's computer software by permitting free access to it.⁴⁰² Following from all those aforementioned reasons, the limited use exception set forth in section 117(a) of the Copyright Act has evolved to a weak defense during the years following.⁴⁰³ Despite the successful raising of section 117(a) as defense in *Vault*, none of the other federal circuits has adopted the Fifth Circuit's approach of section 117(a)(1).⁴⁰⁴ Consequently, *Vault* may be deemed overruled on the application of section 117(a) in the context of intermediate copying. Furthermore, given the fact that most of the software is subject of licensing agreements rather than purchase contracts, the practical relevance of section 117(a) applied to intermediate copying in a reverse engineering context, in particular after *MAI Systems*, may likely be remote. To conclude, reverse engineers who are engaged in possibly objectionable acts of intermediate copying in the course of decompilation have changed their litigation strategy. So, U.S. courts began to make use of the equitable nature⁴⁰⁵ of the new defense⁴⁰⁶ instead invoked in cases of copyright infringement claims—the “fair use” defense.⁴⁰⁷

2. The Fair Use Defense

⁴⁰² Gage, Jr., *supra* note 81, at 188.

⁴⁰³ According to the prevailing view among commentators on U.S. copyright law, *Vault Software* may be deemed overruled as regards the application of section 117(a)(1) to intermediate copyists.

⁴⁰⁴ The Ninth Circuit found it unnecessary to extend section 117(a) to protect uses beyond those intended by the copyright owner. *See Sony*, 203 F.3d 596, at 600 n.1 (refraining from addressing defendant's contention that its copying were within the exception of section 117 of the Copyright Act, while the Ninth Circuit only discussed the fair use defense); *see id.* citing *Sega*, 977 F.2d at 1518 (“Neither the language of the [Copyright] Act nor the law of this circuit supports [defendant] Accolade's argument [that disassembly is authorized by section 117 of the Copyright Act]. [defendant] Accolade's [fair use] argument, however, has merit.”).

⁴⁰⁵ *See Harper & Row Publishers, Inc. v. Nation Enterprises*, 471 U.S. 539, 560 (1985) (characterizing the fair use doctrine as in essence as “an equitable rule of reason”) (citing H.R.REP. No. 1476, 94th Cong., 2d Sess. 65, *reprinted in* 1976 U.S.C.C.A.N. 5659, 5679).

⁴⁰⁶ The three-judge panel in *Sega* held that “consideration of the unique nature of computer object code...is...appropriate as part of the case-by-case equitable 'fair use' analysis authorized by section 107 of the Act.” *See Sega*, 977 F.2d at 1520.

⁴⁰⁷ Donahoe, *supra* note 65, at 111301.

2.1. Section 107 of the Copyright Act

Whereas section 117 of the Copyright Act generally defines acts of copying as being lawful *per se*, section 107 establishes a defense to an otherwise valid claim of copyright infringement. In contrast to *Vault*, jurisprudence has recognized intermediate copying as act of potential copyright infringement. The fact, however, that Congress has not chosen to provide a *per se* exemption to section 106 for decompilation (or disassembly) is not regarded to mean that particular instances of decompilation (or disassembly) may not be subject to the fair use exception.⁴⁰⁸ Therefore, courts turned to section 107 to answer the question whether it is fair use for a user lawfully in possession of a computer program in object code format to decompile that code for the sole purpose of manipulating elements of that code for permissible purposes, when no other readily available substitute mechanism exists for that purpose.

The statute sets out four factors always to be weighed by courts⁴⁰⁹ in determining whether an infringement is fair use:

- (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.⁴¹⁰

⁴⁰⁸ *Sega*, 977 F.2d at 1521.

⁴⁰⁹ The multi-factor test codified in section 107 allows courts flexibility in reaching a decision based on the facts of each case. The four factors are not meant to be exhaustive and no factor is dispositive but all of the statutory factors must be considered in the analysis. *See Campbell v. Acuff-Rose Music, Inc.*, 510 U.S. 569, 577-8 (1994). The intent behind drafting flexibility into the statute was to allow courts to adapt the fair use doctrine to society's evolving needs, especially during times of rapid technological change. *See H.R. REP. NO. 94-1476*, 94th Cong., 2d Sess. 66 (1976).

⁴¹⁰ 17 U.S.C. § 107 (1992) says in full text:

“Notwithstanding the provisions of sections 106 and 106A, the fair use of a copyrighted work, [...], 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 determining whether the use made of a work in any particular case is a fair use the factors to be considered shall include—

Under a traditional analysis, none of the factors above would support a finding of fair use for decompilation of computer programs. The purpose and character of the use is competitive, private commercial gain. Second, the nature of a computer program is composed of creative expression of the programmer, whose substantial investments are made in hopes of financial return. Third, the person who decompiles software must make an absolute copy of the entire program, duplicating both protected and unprotected expression in order to access the underlying utilitarian concepts. Fourth, allowing competitors to infringe a copyright owner's computer program could destroy the software owner's potential market by decreasing the demand for the author's work. In other words, secondcomers on the market would reap the benefits of the author's work without absorbing the R&D costs, thus destroying the author's incentive to produce.

Upon amendment of section 107 in October of 1992,⁴¹¹ Senator Simon, who introduced the bill, apparently stuck with the traditional fair use analysis when he stated that "this bill is not intended to provide a fair use access to [invisible and thus unpublished computer codes] through decompilation."⁴¹² Also the majority report by the Senate Judiciary Committee confirmed that "nothing in the bill is intended in any way to broaden fair use of unpublished computer programs."⁴¹³ However, case law has proved not to abide with the legislators' intent.

Still before the enactment of the amendment, on Sept. 10, 1992, the federal circuit, in *Atari Games Corp. v. Nintendo of America Inc.*,⁴¹⁴ first faced the reverse engineering question and with its judgment launched a revolution in the legal as well as the software world.

-
- (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 fact that a work is unpublished shall not itself bar a finding of fair use if such finding is made upon consideration of all the above factors."

⁴¹¹ See 17 U.S.C. § 107, as amended Oct. 24, 1992, Pub. L. 102-492, 106 Stat. 3145. The fair use provision was amended with respect to unpublished works—in that an unpublished work "shall not itself bar a finding of fair use if such finding is made upon consideration of all the above factors."

⁴¹² See 137 CONG.REC. S5648 (daily ed. May 9, 1991).

⁴¹³ See 137 CONG.REC. S13925 (daily ed. Oct. 27, 1991).

⁴¹⁴ 975 F.2d 832, 836 (Fed. Cir. 1992) [hereinafter Atari].

2.2. *Atari Games Corp. v. Nintendo of America Inc.*

2.2.1. The Facts

Nintendo of America, Inc. (“Nintendo”) manufactured a video game console and marketed game cartridges for use in the console. Both devices were equipped with the copyrighted software program “10NES”, a combination of “lock” software embedded into a chip in the game console, and “key” software in each Nintendo game cartridge. The lock and key send synchronized encoded data streams back and forth which unlock the console when an authorized game is inserted. When an unauthorized game is inserted, the console remains locked, thus preventing rival game manufacturers from designing competitive Nintendo-compatible games without receiving keys from Nintendo.⁴¹⁵

Atari Games Corporation (“Atari”), one such competitor of Nintendo, began efforts to replicate the 10NES program through examination of the chips embodying the computer code, hoping to make its own games compatible with Nintendo’s more popular console. When Atari’s attempts failed to dissect (i.e. decompile) the microchips and, subsequently, to reverse engineer the Nintendo code,⁴¹⁶ it entered into a license agreement with Nintendo, which terms strictly controlled Atari’s access to Nintendo’s “lock and key” code. Thereupon, Atari’s agents procured a deposit copy of Nintendo’s source code by misrepresenting to the U.S. Copyright Office that Atari was involved in actual litigation with Nintendo and that the copy would be used in connection with that litigation.⁴¹⁷ Using this copy of the source code, Atari successfully created its own “Rabbit” program, to mimic the appropriate signal that unlocks the console control.⁴¹⁸ Although the Rabbit was developed using a different microprocessor and different computer language, it sent

⁴¹⁵ *Atari*, 975 F.2d at 832, 836.

⁴¹⁶ Atari made two intermediate copies of the program while attempting to reverse engineer the 10NES code: First, Atari stripped the 10NES and copied portions of the object code, and, second, Atari made photocopies and hardcopies of the 10NES object code and entered the object code into a computer. *See* 975 F.2d at 836, 842.

⁴¹⁷ *Id.* at 841-2.

⁴¹⁸ Atari took two actions in order to overcome the “lock and key” device: It performed chemical peeling, transcribing zeroes and ones therefrom via microscopic examination and, to assist it in that process, it obtained copies of parts of the Nintendo source code from the Copyright Office under false pretenses to correct errors in the transcription. *Id.* at 836.

data streams to the Nintendo console identical to those sent by Nintendo games. Atari was thereby able to sell game cartridges to consumers without breaching the strict licensing controls imposed by Nintendo. Since Nintendo could no longer stop Atari from building Nintendo-compatible games technologically, it was determined to stop them legally. Nintendo filed suit, inter alia, for copyright infringement of the 10NES security program.⁴¹⁹

2.2.2. The Decision

On appeal, the Federal Circuit upheld the injunction enjoining the production of Atari's Rabbit program and found that Atari infringed Nintendo's copyright by obtaining an unauthorized copy of the program.⁴²⁰ In order to use the fair use defense in a case involving decompilation the copier must begin with a legitimate copy. In other words, Atari could not use the unauthorized copy of the transcript obtained from the Copyright Office as a successful basis for its fair use defense.⁴²¹ Because Atari did not need to employ acts of decompilation since it anyway fraudulently obtained a copy of Nintendo's source code from the Copyright Office after unsuccessful attempts to decompile Nintendo's code, the court discussed decompilation merely as an alternative ground. In assessing the legality of software decompilation in cases using authorized copies the Federal Circuit *in dictum* made Atari's conduct subject to the multi-factor test of section 107 of the Copyright Act.

2.2.3. Fair Use Analysis

In applying Ninth Circuit law, the Federal Circuit held decompilation of object code to discover the unprotectible ideas in a computer program non-infringing fair use.⁴²² The court underwent a fairly cursory analysis of the four factors upon which it based its holding. Under the first factor, the purpose and character of making reproductions of computer code, intermediate copying as occurred in the course of decompilation does not extend to cover Atari's commercial exploitation.⁴²³ In addressing the second factor, the *Atari* court concluded that "when the nature of a

⁴¹⁹ *Id.* at 835-7.

⁴²⁰ *Id.* at 835.

⁴²¹ *Id.* at 845.

⁴²² *Id.* at 844.

⁴²³ *Id.* at 843.

work requires intermediate copying to understand the ideas and processes in a copyrighted work, that nature supports a fair use for intermediate copying.”⁴²⁴ The third and fourth factors—but not identified as such—were dealt with inextricably.⁴²⁵ Essentially, in its fair use analysis the court limited Atari’s conduct to copying only those portions of the 10NES program required to send the correct data sequence to the Nintendo console.⁴²⁶

2.2.4. The “Legitimate Owner” Requirement

Article 6(1)(a) of the EU Software Directive requires the lawfulness of users of a computer program as condition on the legitimacy of decompiling that program.⁴²⁷ Across the Atlantic, in the *Atari* case, the court found *in dictum* that copyright law could not prevent one from delving into a lawfully obtained copy of a program to ascertain its idea and functional elements.⁴²⁸ Since Atari had purloined the copy of 10NES source code it was precluded from relying on a fair use defense. However, there are two lines of argumentation that might weaken such requirement resembling the EU model prescribing that the copy of the target program must be lawfully acquired in order for acts of decompilation to be permissible. First, none of the courts opining in follow-up decompilation cases did reiterate the condition.⁴²⁹ Second, the CONTU Report was adopted by Congress verbatim with one change: The final version grants “owners”, as opposed to “rightful possessors” a limited right to copy and adapt their software.⁴³⁰ Thus, the current wording of section 107 of the Copyright Act seems not to bother with the lawfulness of users of a computer program. The reason that there is no legislative history as to why this change was adopted⁴³¹ is just another indicator for the doubtful impact of such requirement under U.S. case law.

⁴²⁴ *Id.*

⁴²⁵ *But cf.* Gage, Jr., *supra* note 25, at 195-6 (denying that the *Atari* court at all analyzed three of the four fair use factors).

⁴²⁶ *Atari*, 975 F.2d, at 843.

⁴²⁷ EU Software Directive, *supra* note 27, art. 6(1)(a), at 45 (reading that “these acts are performed by the licensee or any other person having a right to use a copy of a program, or on their behalf by a person authorized to do so”).

⁴²⁸ *Atari*, 975 F.2d at 843.

⁴²⁹ The facts in those cases, however, show that the defendant reverse engineers obtained lawful copies of the target computer software.

⁴³⁰ *See, e.g., Vault*, 847 F.2d 255 at 261.

⁴³¹ Bayha, *supra* note 8, at 186 n.15.

All in all, given the highly unusual facts underlying the *Atari* ruling, its precedential authority appears of highly limited relevance.⁴³² Hence, the legal as well as the computer world were waiting with eager anticipation of the courts' further steps in the software decompilation battle.

2.3. *Sega Enterprises, Ltd. v. Accolade, Inc.*

Also in 1992, within a month of each other, another federal court, the Ninth Circuit, held that decompilation (here in the form of disassembly) of copyrighted object code is fair use.⁴³³ Coincidentally, the second case involved Nintendo's giant competitor of the U.S. computer video game industry, Sega Enterprises, Ltd. ("Sega").

2.3.1. The Facts

Sega manufactured the Genesis video game console containing a "lock-out" feature. Through this device, the cartridge would not run unless the console found a security code present in a video game cartridge—consisting of about 25 bytes of data.

Accolade, Inc. ("Accolade") purchased a Genesis console and three Sega game cartridges and wired a decompiler into the console. Accolade engineers studied the printouts produced by the decompiler, loaded the decompiled object code into a computer and experimented to determine the interface specifications for the Genesis console. Accolade then developed a manual containing the functional descriptions of the interface requirements, but it did not contain the actual source code found in the Sega program itself. Subsequently, Accolade engineers decompiled a 25 bytes header file which was based upon the information concerning interface specifications for the Genesis console that was contained in its development manual into Accolade's own video game programs, thus enabling them to operate on the Genesis console.⁴³⁴ Sega filed suit, complaining that

⁴³² See 4 NIMMER ON COPYRIGHT, *supra* note 15, § 13.05[D][4], at 13-230.22; see also Gage, Jr., *supra* note 81, at 195-6 (finding fault with the court's "tenuous" legal fair use analysis).

⁴³³ *Sega Enterprises Ltd. v. Accolade, Inc.*, 977 F.2d 1510 (9th Cir. 1992) [hereinafter *Sega*].

⁴³⁴ 977 F.2d at 1514-5.

Accolade violated Sega's copyright by decompiling and reverse engineering the Sega software as contained in its Genesis console in a clean room procedure.⁴³⁵

2.3.2. The Decision

Accolade contended that intermediate copying of computer object code is fair use and thus does not constitute copyright infringement. The district court rejected Accolade's argument by reasoning that Accolade had decompiled Sega's code for a commercial purpose, and that Sega had likely lost sales of its games. Furthermore, there would have been alternatives to decompilation to be used by Accolade in order to study the functional requirements for Genesis compatibility.⁴³⁶

On appeal, the Ninth Circuit underwent a fair use analysis resulting in the conclusion that "where [decompilation] is the only way to gain access to the ideas and functional elements embodied in a copyrighted computer program and where there is a legitimate reason for seeking such access, [decompilation] is a fair use of the copyrighted work, as a matter of law."⁴³⁷

2.3.3. Fair Use Analysis

Starting its analysis with the first factor, the Ninth Circuit considered the commercial nature of Accolade's direct copying as an intermediate use only. The court treated the issue of copying as a matter of degree. Accordingly, any commercial exploitation was merely indirect or derivative.⁴³⁸ As regards the purpose of the allegedly infringing conduct, the court made the act of intermediate copying itself subject to the analysis, and not the competing end-product as result of intermediate copying. The direct purpose was to study the functional requirements in Sega's code needed for compatibility with the Genesis console. Focusing on Accolade's own independently designed, non-infringing video game programs as the result of its reverse engineering process the court found that Accolade's decompilation of Sega's code has led to a "growth in creative expression, based on the dissemination of [more] creative works [of authorship]",

⁴³⁵ *Id.* at 1516.

⁴³⁶ *Id.* at 1517.

⁴³⁷ *Id.* at 1527-8.

⁴³⁸ *Id.* at 1522.

and thereby promoting the goals of copyright law.⁴³⁹ Ultimately, the court concluded that the commercial aspect of Accolade's use, i.e. offering its reverse-engineered Genesis-compatible video games for sale on the market, is merely of minimal significance because any of Sega's "minor economic loss" was outweighed by public benefit.⁴⁴⁰ Given the "essentially non-exploitative purpose" the presumption that commercial uses are unfair was deemed rebutted.⁴⁴¹

Turning directly to the fourth factor, the court inquired whether Accolade's intermediate copying, assuming that the challenged use would become widespread, would adversely affect the potential market for Sega's video games. The court, again, seemed to draw a line between the process of decompilation and the subsequent producing of independent creative works. By focusing on the crucial stage of Accolade's direct copying there was found no direct competition regarding the sales of the "lock and key" software components. In other words, since Accolade's intermediate copies have never entered the marketplace, neither was the marketability of Sega's copyrighted games impaired nor the value of Sega's copyrighted expression therein diminished.⁴⁴² At the video game market, however, Accolade would appear as Sega's competitor and would likely cause loss of some sales of Sega's video games. Since neither were Accolade's own games substantially similar to those ones of Sega,⁴⁴³ nor should a *de facto* monopoly on the video game market be conferred upon Sega by making it impossible for others to compete,⁴⁴⁴ the Ninth Circuit held the fourth factor to tip against Sega. However, despite the court's knowledge that other developers had already purchased Sega licenses, the court ignored to take into account that Sega would probably be exposed a serious, potential risk of losing licensing opportunities. Instead of purchasing from Sega, those wanting to develop Genesis-compatible videogames

⁴³⁹ *Id.* at 1523.

⁴⁴⁰ *Id.* at 1524.

⁴⁴¹ *Id.*

⁴⁴² See II GOLDSTEIN ON COPYRIGHT, *supra* note 12, § 7.2.1.4.c, at 7:57; see also 4 NIMMER ON COPYRIGHT, *supra* note 15, § 13.05[D][4], at 13-231.

⁴⁴³ 977 F.2d at 1523 (stating that "it is the characteristics of the game program as experienced by the user that determine the program's commercial success" and that there could not be assumed a threat of effectively usurping the market for Sega's video games by supplanting its games because potential consumers in this field "might be expected to buy more than once game").

⁴⁴⁴ *Id.* at 1523-4, 1527 ("[A]n attempt to monopolize the market by making it impossible for others to compete runs counter to the statutory purpose of promoting creative expression and cannot constitute a strong equitable basis for resisting the invocation of the fair use doctrine.").

can simply decompile Sega's code while being protected under the fair use shield.⁴⁴⁵

As to the second factor, the Ninth Circuit found that Accolade had no viable alternative to decompiling Sega's computer code in order to discover the functional interface specifications for a game compatible with Sega's product.⁴⁴⁶ Computer programs as essentially utilitarian works contain numerous unprotected elements dictated by function, considerations of efficiency and compatibility requirements. Based on the evidence, the court ruled that the object code in Sega's games contained unprotected elements, including interface specifications, which could not be examined by humans without decompiling the target code with the consequence of making copies thereof.⁴⁴⁷ The court thus accorded video games a lower degree of copyright protection and concluded that this statutory factor likewise weighed in favor of Accolade.⁴⁴⁸

In applying the second statutory factor, the court found that Accolade copied the protected Sega code in its entirety. While this factor thus weighed against Accolade, the court stressed that copying of an entire work does not by and of itself necessarily preclude a finding of fair use.⁴⁴⁹

2.3.4. Conclusion

At the end of the day, the Ninth Circuit held Accolade's intermediate copying to be fair use as a matter of law, thereby establishing the *Sega* rule. Two conditions must be satisfied under the *Sega* rule for decompilation (or disassembly)—generally, for making intermediate copies in the process of reverse engineering—to be seen as a fair use. First, making intermediate copies through decompilation (or disassembly) must be “the only way to gain access to the ideas and functional elements embodied in a copyrighted computer program”. Second, there must be a “legitimate reason” for gaining access.⁴⁵⁰ The *Sega* ruling protects only acts of decompilation (or disassembly) itself. To the extent that the copier markets an ultimate product

⁴⁴⁵ See generally John A. Williams, *Can Reverse Engineering of Software Ever Be Fair Use? Application of Campbell's "Transformative Use" Concept*, 71 WASH. L. REV. 255, 271-2 (1996).

⁴⁴⁶ 977 F.2d at 1526.

⁴⁴⁷ *Id.* at 1525.

⁴⁴⁸ *Id.* at 1526.

⁴⁴⁹ *Id.* at 1525-6.

⁴⁵⁰ *Id.* at 1514, 1527.

embodying copyrighted protectible expression, an infringement claim may still lie.⁴⁵¹

The two prongs of the *Sega* rule resemble the indispensability criterion and the limited purpose requirement as laid down in Article 6 of the EU Software Directive. Like Article 6, the *Sega* rule solely focuses on the intermediate copying itself in the course of decompiling the target program, while the subsequent step of creating a new program based on the information decompiled is subject to an independent legal inquiry. However, the *Sega* rule is even more undefined in each of its prongs than Article 6. Does the *Sega* ruling measure the “only way” criterion in terms of a mere technical test or do also efficiency considerations count? In contrast to Article 6, the “legitimate reason” requirement is not limited to the purpose of interoperability with the consequence that also software decompilation for research purposes can be subsumed under the *Sega* rule.

2.3.5. Unpublished Source Code

In discussing the nature of *Sega*’s copyrighted computer program under the second prong of the fair use analysis, the hybrid nature of computer programs in general was taken into account. The court generously explained that computer programs are typically distributed for public use in object code form implemented in a silicon chip or on a floppy disk. As a consequence of the object code being invisible for its users, humans cannot gain access to the unprotected ideas and functional concepts contained in object code without an intermediate step, such as decompilation.⁴⁵²

Sega’s legal counsels, however, did not take a closer look—and neither did the Ninth Circuit—at the interrelated structure of object and source code. Albeit computer programs are, in essence, utilitarian works it, nevertheless, is the programmer’s choice of program structure and design that makes a computer program creative and idiosyncratic. These creative choices are expressed in specialized alphanumeric languages, the source code. In order to operate a computer, source code must be transformed into computer readable form using only zeros and ones, the object code. That means that the protected creative expression is not lost simply because it is transformed into another language. *Sega*

⁴⁵¹ See *id.* at 1528; see also 4 NIMMER ON COPYRIGHT, *supra* note 15, § 13.05[D][4], at 13-232.

⁴⁵² See *id.* at 1525.

kept its source code secret but published its object code. In case law thus far unpublished works, like Sega's secret source code, enjoyed greater immunity from the fair use defense.⁴⁵³ Consequently, one might question how had the Ninth Circuit handled the analysis of the second fair use factor would Sega have drawn its attention to the protected elements contained in the unpublished source code rather than in the published object code. In spite of the interrelated structure of object and source code, the Ninth Circuit in focusing on the intermediate copying itself would probably have achieved the same outcome since decompilation means the reading the zeros and ones of the object code that are produced while the program is being run, storing them in computer memory, and translating them into source code. Sega's line of argumentation as to take account of the protected elements in its source code would likely fail as well given the fact that decompilation involves copying from the published object code⁴⁵⁴ and not the unpublished source code. No such discussion took place in the drafting stage of the EU Software Directive. It was well recognized the fact that computer programs normally are distributed in object code form but without further distinguishing between object and source code in terms of legal consequences in the process of decompilation.

2.4. *Sony Computer Entertainment, Inc. v. Connectix Corp.*

2.4.1. The Facts

In 2000, the Ninth Circuit revisited the issue of decompilation under the fair use concept.⁴⁵⁵ The Ninth Circuit found that three of the statutory fair use factors favored the defendant Connectix Corp. ("Connectix"), and thus determined that Connectix's intermediate copying of the Sony BIOS during the course of decompilation was a fair use, as a matter of law.⁴⁵⁶

Plaintiff Sony Computer Entertainment, Inc. ("Sony") developed and marketed a video entertainment console, known as the Sony PlayStation, which connects to a

⁴⁵³ See, e.g., *Harper & Row Publishers, Inc., v. Nation Enterprises.*, 471 U.S. 539, 552-3 (1985); *Salinger v. Random House, Inc.*, 811 F.2d 90, 97 (2d. Cir.), *cert. denied*, 484 U.S. 890 (1987).

⁴⁵⁴ Cf. II GOLDSTEIN ON COPYRIGHT, *supra* note 12, § 7.2.1.4.c, at 7:55-6 n.139.

⁴⁵⁵ *Sony v. Connectix*, 203 F.3d 596 (9th Cir. 2000), *cert. denied*, 531 U.S. 871 (2000) [hereinafter Sony].

⁴⁵⁶ See *Sony*, 203 F.3d at 608.

television set, and software games embedded on compact discs. By loading the compact discs into the PlayStation, users of Sony's system are able to play the games on a television. The PlayStation comprised both hardware components and software written on a read-only memory (ROM) chip. Sony owned the copyright in the basic input-output system (BIOS), the operating system software, for its PlayStation video game system. Connectix was in the process of developing an emulated software, its Virtual Game System (VGS), that would allow PlayStation games to be played on a conventional personal computer instead of on a television connected to a PlayStation console.⁴⁵⁷ To ensure compatibility between VGS and the PlayStation games, Connectix had to decompile the PlayStation. In the process of developing software which interacts with the PlayStation BIOS, Connectix engineers had to load the BIOS into a computer and run it repeatedly. The repeated running of the BIOS caused the making of numerous temporary copies of the entire BIOS in the computer's memory for examination. Based on its review of the PlayStation BIOS, Connectix then developed its own BIOS code and included it in the final Connectix product. However, the final Connectix product contained none of Sony's copyrighted computer code.⁴⁵⁸

2.4.2. The Decision

Sony asserted that these copies infringed its copyright in the PlayStation BIOS.⁴⁵⁹ Relying on the Ninth Circuit's *Sega* decision, Connectix responded that the intermediate copies were excused under the fair use doctrine because they were necessary for disclosure of elements unprotected by Sony's copyright—the PlayStation BIOS's interface specifications.⁴⁶⁰ Sony countered by arguing that Connectix could have avoided the making of copies of its BIOS had it followed a different development process. Connectix could have developed its own BIOS at the beginning, and used that BIOS in the development of its own interoperable software, rather than use the PlayStation BIOS in the development of the

⁴⁵⁷ A consumer can load the VGS software onto a computer, load a PlayStation game into the computer's CD-ROM drive, and play the PlayStation game. The VGS software's purpose is to emulate both the hardware and software components of the Sony console. *See Sony*, 203 F.3d at 599.

⁴⁵⁸ 203 F.3d at 601.

⁴⁵⁹ *Id.*

⁴⁶⁰ *Id.* at 602, 604.

interoperable software, and then develop its own BIOS.⁴⁶¹ The Ninth Circuit rejected Sony's argument:

Even if we were inclined to supervise the engineering solutions of software companies in minute detail, and we are not, our application of the copyright law would not turn on such a distinction. Such a rule could be easily manipulated. More important, the rule urged by Sony would require that a software engineer, faced with two engineering solutions that each requires [viz] intermediate copying of protected and unprotected material, often follow the *least efficient solution*. ... This is precisely the kind of "wasted effort that the proscription against the copyright of ideas and facts ... [is] designed to prevent." Such an approach would erect an artificial hurdle in the way of the public's access to the ideas contained within copyrighted software programs.⁴⁶²

In considering the general rule set forth in *Sega* according to which decompilation is, as a matter of law, a fair use of the copyrighted target program "where [decompilation] is the only way to gain access to the ideas and functional elements embodied in a copyrighted computer program and where there is legitimate reason for seeking such access."⁴⁶³ The *Sony* case appears to hold to the same core dogma.⁴⁶⁴ However, the *Sony* analysis of the fair use doctrine was vastly different in the way how to reach that outcome.

2.4.3. How "Transformative" is the Sony Ruling Compared to the Sega Concept?

In considering the purpose and character of the infringing use, the court focused on the question of whether the VGS was "transformative"⁴⁶⁵ of the PlayStation—an issue not discussed in *Sega*. For the court, particularly, the fact that Connectix's final product, the VGS, did not contain or otherwise infringe any of Sony's copyrighted code was crucial in answering this question in the affirmative.⁴⁶⁶

Eight years after *Sega*, Judge William Canby Jr., one of the judges who sat on the panel that decided *Sega*, authored the *Sony* opinion. In both cases, it was the

⁴⁶¹ *Id.* at 605.

⁴⁶² *Sony*, 203 F.3d at 605 (quoting *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 354 (1991) (emphasis added)).

⁴⁶³ *Sega*, 977 F.2d at 1527-8.

⁴⁶⁴ *Id.* at 609-10 (holding that "Connectix's reverse engineering of the Sony BIOS extracted from a Sony PlayStation console purchased by Connectix engineers is protected as a fair use").

⁴⁶⁵ *Id.* at 606 (quoting *Campbell*, 510 U.S. at 579, in identifying whether Connectix's VGS "merely supersedes the objects of the original creation, or instead adds something new, with a further purpose or different character, altering the first with new expression, meaning, or message").

⁴⁶⁶ *Id.* at 606-7.

intermediate copies—those that were generated in the course of decompilation—that were alleged to be infringing works, not the final works. In *Sega*, the final work was a new game cartridge that could be used on the Sega Genesis game console. In *Sony*, the final work was an emulation program that allowed PlayStation games to be played on a Macintosh computer; in other words, an alternative platform for Sony games. In both cases, the intermediate works were created as an essential step in the creation of a final, non-infringing product.

In the copyright case *Campbell v. Acuff-Rose Music Inc.*,⁴⁶⁷ the U.S. Supreme Court ruled that the more a new work is “transformative” the less the other purposes and characters of the use, such as commercialism, are significant.⁴⁶⁸

Taking its cues from *Campbell*, the analysis of the first factor in the *Sony* case centered on the extent to which Connectix’s final product was transformative.⁴⁶⁹

The court’s analysis in *Sega*, however, focused on the purpose and character of the intermediate copies.⁴⁷⁰ Only as an afterthought did the court note that the final use promoted the goals of the Copyright Act by encouraging creative expression in the form of “independently designed video game programs offered for use with the Genesis console.”⁴⁷¹ Although some language from *Sega* was quoted, no direct reference was made to *Sega*’s holding that the purpose of the intermediate copies was to identify the unprotectible expression.⁴⁷² In contrast to *Sega*, the central question for the *Sony* court seemed to be whether the final work “merely supersedes the objects of the original creation, or instead adds something new, with a further purpose or different character, altering the first with new expression, meaning, or message...in other words, whether and to what extent the new work is

⁴⁶⁷ *Campbell v. Acuff-Rose Music Inc.*, 510 U.S. 569 (1994) [hereinafter *Campbell*].

⁴⁶⁸ *Campbell*, 510 U.S. at 579 (Justice Story focusing on whether the new work merely supersedes the objectives of the original creation, or instead adds something new with a further purpose or different character, and thereby transforming the underlying expression, meaning or message).

⁴⁶⁹ *Sony*, 203 F.3d at 606.

⁴⁷⁰ *Sega*, 977 F.2d at 1522-3 (discussing the ultimate purpose of Accolade’s acts of disassembling Sega’s computer code).

⁴⁷¹ 977 F.2d at 1523.

⁴⁷² The *Sega* court buttressed its holding by stating that the purpose behind the copies was to separate the protectible from the non-protectible components of a work. Accolade was held to have copied “Sega’s code for a legitimate, essentially non-exploitive purpose, and that the [indirect or derivative] commercial aspect of its use can best be described as of minimal significance.” See *Sega*, 977 F.2d at 1522-3.

‘transformative.’”⁴⁷³ The *Sony* ruling undermined the “legitimate reason” requirement set forth in the *Sega* rule by reasoning that a final product which does not contain any code of the original product is transformative. A transformative product does not supplant the original product and thus does not cause a substantially adverse impact on the potential market of the original.⁴⁷⁴ To conclude succinctly: Same Circuit, different focus.

If the *Sony* rule is the new standard, then what if a final work is actually intended to supersede the objects of the original creation? What if the outcome of decompiling a computer program would be a look-alike clone product, though, with no substantially similar computer code? What if a function could be coded in a different manner, but it would not be quite as fast or quite as concise? Under the *Sega* rule, the question would have been whether the decompilation was done in order to observe the non-protectible elements of the program. Now, *Sony* directs us to look to the final product and ask: How does the final product alter the objects of the original creation? *Sony* itself provides an answer to the question. When discussing the transformative nature of Connectix’s product, the court identified the creation of the VGS as a new platform that provides “opportunities for game play in new environments, specifically anywhere a Sony PlayStation console and television are not available, but a computer with a CD-ROM drive is.”⁴⁷⁵ The thrust of the court’s analysis reads as follows:

More important, the [Connectix program] itself is a wholly new product, notwithstanding the similarity of uses and functions between the Sony PlayStation and [Connectix’s program]. ...Sony does not claim that [Connectix’s program] itself contains object code that infringes Sony’s copyright. We are therefore at a loss to see how Connectix’s drafting of entirely new object code for its...program could not be transformative, despite the similarities in function and screen output.⁴⁷⁶

With the focus on the purpose and character of the final use it is a significant departure from the *Sega* precedent. Although the language clearly supports a finding that all non-infringing final works are *per se* transformative, debates in

⁴⁷³ 203 F.3d at 606.

⁴⁷⁴ Rothman, *supra* note 97, at 6-7 (showing the Ninth Circuit’s line of reasoning which leaves little room for any meaningful application of the *Sega* rule, respectively, the fair use defense to be applied as a significant departure from *Sega*).

⁴⁷⁵ *Sony*, 203 F.3d at 606.

⁴⁷⁶ *Id.*

courtrooms are likely to move from whether a final work is infringing to whether a final work is transformative. The general rule that decompilation is a fair use as a matter of law is, therefore, threatened to be diluted.

Merely stating that the final work was transformative was all the proof the *Sony* court needed to hold that this factor was in favor of Connectix. Here both courts reached the same outcome. However, there may well be cases where the distinction between the purpose and character of intermediate works and the purpose and character of final works may prove significant with respect to their outcomes. For example, what if Corel decompiled Microsoft Word in order to enhance its own word processor, WordPerfect? In the hypothetical, the outcome of this factor may change, depending on whether a *Sega* or *Sony* analysis is followed. Although proof that sales of WordPerfect displace sales of Microsoft Word would probably sway the *Sega* court to an insignificant extent, the decisive inquiry of the court would be whether the final product infringes Microsoft's copyright in its Word program. If the end product contains any of the original work's protected elements, the reverse engineer will move from the safety of the intermediate copying category into direct copying. This will likely result in the fair use factors shifting towards a finding of infringement.⁴⁷⁷ Would the case be litigated in front of the *Sony* court, the transformative nature of Corel's improved WordPerfect would determine the outcome irrespective of the amount of copyrighted expression as contained in the Microsoft Word and re-used in the new WordPerfect.

The Ninth Circuit goes a step farther than the European legislators did. Whilst in Europe the debate was about whether decompilation as a preliminary step in the development of a competing end-product shall be permissible or not, permissibility of acts of decompilation under the *Sony* analysis seems to turn on the transformative element of the new program. Presumably the new program can be of a higher development and thus be transformative of the target program with the result to be condoned from copyright infringement, even though the new program may well become a competing product to the target program at the market.

⁴⁷⁷ See *Sony*, 203 F.3d at 606-7; see also Donahoe, *supra* note 65, at 111301 (exemplifying a company that develops a clone of a competitor's product via decompilation will be likely determined to be infringing even where the copyrighted software was only used in an intermediate fashion).

The fourth statutory factor deals with the effect on the potential market for the copyrighted work.⁴⁷⁸ Beyond this presumption, the court did not attempt an in-depth analysis of the true market effect of the VGS, as an alternative platform for Sony games, on sales of Sony's platform.⁴⁷⁹

In *Sega*, the analysis was fact-specific. Consumers usually buy several game cartridges, even in the same genre. The copyrightable elements in each game dictate their market success, and none of these elements was infringed by Accolade. The intermediate copies merely permitted Accolade to become a market competitor, something Sega could not legally prevent. After considering these factors, the court found that the effect on the potential market was negligible, in that any commercial exploitation from the intermediate use was characterized as "indirect or derivative".⁴⁸⁰ The analysis in *Sony* was more generic. By citing *Campbell*, the court put the possible effect on the market of the target product in relation with the transformative nature of the decompiled product and concluded that transformative works are not likely to have any adverse impact on the copyright holder's potential market.

2.5. *The Indispensability Criterion*

The *Sega* rule affords a finding of two factors in order to establish decompilation as fair use. Under the first prong, decompilation must be the only way to gain access to unprotected ideas and functional elements of computer programs.⁴⁸¹ What under the *Sega* rule is flatly required as the "only way", poses a variety of questions in different directions, all of which have already been dealt with in Europe likewise.

2.5.1. Indispensability under the *Sega* Rule

⁴⁷⁸ *See id.*, at 607 (reasoning that "[w]hereas a work that merely supplants or supersedes another is likely to cause a substantially adverse impact on the potential market of the original, a transformative work is less likely to do so").

⁴⁷⁹ *See id.* (with the effect of avoiding to draw a line between acceptable and non-acceptable levels of economic loss, the Ninth Circuit held in a superficial manner that Connectix's PC-based platform for Sony games might reduce sales of a television-based platform and thus causing at least "some"—in the sense of acceptable—economic loss); *see also* Rothman, *supra* note 97, at 7.

⁴⁸⁰ *Sega*, 977 F.2d at 1522.

⁴⁸¹ 977 F.2d at 1514, 1527.

From a technical point of view, the Ninth Circuit distinguished the facts in *Sega* from those in *Atari*.⁴⁸² Whereas in *Sega* the target code to unlock the Genesis console consisted merely of 20 bytes of initialization code plus the letter S-E-G-A, the *Atari* court found that there was a multitude of different ways to unlock the Nintendo console.⁴⁸³ The *Sega* court seems implicitly to say that technically the process of decompilation was the only measure at hand to uncover the exact Sega 20 byte security code required to achieve compatibility. Any other way, irrespective of whether such a means would have been less infringing or economically more efficient, was expounded. Thus, the act of decompilation was held fair use without considering the availability or relevance of any other way to gain access to the interface information Accolade required.

After all, the first prong of the *Sega* rule was to be understood in a narrow meaning as decompilation must be the “only” resort to access of interface information in order to be permissible.

2.5.2. Indispensability under the Sony Rule

The *Sony* court chose to use the first factor, the nature of the copyrighted work, to hold that the manner of decompilation is unimportant in the court’s analysis. In this context, Sony argued that Connectix could have developed the VGS system by first decompiling the PlayStation BIOS in order to study the unprotected interface specifications and to create its own non-infringing BIOS, which would have required the making of only a single copy, and then using its own BIOS to emulate the hardware components of the PlayStation. Instead, Connectix chose to first emulate the hardware through a process of black-box reverse engineering, which required Connectix to make numerous, repeated, daily copies of the Sony BIOS. Sony argued that Connectix allegedly had been acted beyond the “necessity” required under the *Sega* rule.⁴⁸⁴ In rejecting Sony’s argument, the Ninth Circuit stated that the actual number of intermediate copies that are made can depend on whether the code is analyzed as a whole or section-by-section, how often the

⁴⁸² *Id.*, at 1524 n.7.

⁴⁸³ *See Atari*, 975 F.2d at 839-40.

⁴⁸⁴ *Sony*, 203 F.3d at 604.

decompiler is run and whether the decompiled code is saved on a hard drive.⁴⁸⁵ Such details are left up to the engineers and will not be second-guessed by the court.⁴⁸⁶ Given the court's reasoning it appears that the indispensability criterion of the *Sega* rule is to be interpreted in efficiency terms rather than from a rigid technical point of view as it seems to be the case under Article 6 of the EU Software Directive.

In *Sony*, the Ninth Circuit relied on the *Sega* rule as precedent.⁴⁸⁷ But the facts in *Sony* differ technically from those in *Sega* so that the Ninth Circuit in *Sony* applied the *Sega* rule in the meaning of an efficiency analysis. The court, first, reviewed whether Connectix's copying of the PlayStation BIOS was necessary in terms of fair use.⁴⁸⁸ Because object code cannot be read by humans, the court found no alternative means by which one may gain access to the unprotected elements of a computer program but decompiling the target code.⁴⁸⁹ Yet the court went a step further and turned to the question whether the methods by which Connectix decompiled the PlayStation BIOS were necessary to gain access to the unprotected functional elements within the program.⁴⁹⁰ The Ninth Circuit itself enumerated and discussed the following methods of reverse engineering—thereby providing for an order of reverse engineering means that the European legislators never dared to discuss: (1) reading about the program, (2) the so-called “black-box reverse engineering”,⁴⁹¹ and (3) decompilation of object code into source code. The first method was discovered as the least effective one⁴⁹² and as no option at Connectix's disposal because Sony did not make information about its PlayStation accessible.⁴⁹³

⁴⁸⁵ The same question was not presented to the *Sega* court.

⁴⁸⁶ 203 F.3d at 605.

⁴⁸⁷ *Id.*, at 602.

⁴⁸⁸ *Id.*, at 603.

⁴⁸⁹ *Id.* (reasoning that the fact that the Sony BIOS is an internal operating system that does not produce a visible output to reflect its functioning compelled Connectix to gain access to the functional elements of the Sony BIOS through a form of reverse engineering that required (intermediate) copying the Sony BIOS onto a computer).

⁴⁹⁰ *Id.*

⁴⁹¹ A Computer program is input into a computer and the manner in which the computer functions and the output which is generated is observed. *See Bayha, supra* note 8, at 179; *see also Secure Services Tech. v. Time and Space Processing*, 722 F. Supp. 1354, 1361 n.17 (E.D. Va. 1989) (characterizing black box reverse engineering as reverse engineering “in its simplest form”, and upholding the legitimacy of the use of such process to achieve interoperability).

⁴⁹² *See also Johnson-Laird, supra* note 7, at 846 (1994) (stating that individual software manuals often misdescribe the real product).

⁴⁹³ *Sony* 203 F.3d at 599-601.

Additionally, Connectix requested in vain technical assistance from Sony to complete the development of the VGS.⁴⁹⁴ Thereupon Connectix employed methods (2) and (3)—both involve copying the target program in the computer’s memory.⁴⁹⁵ Although the Ninth Circuit expressly announced to prefer neither to another as a matter of copyright law, the holding, nonetheless, reads to sympathize with Connectix as it got primarily engaged in a less intrusive black-box analysis,⁴⁹⁶ and employed decompilation only at a second stage,⁴⁹⁷ when observing the functional aspects of the PlayStation BIOS in an emulated had proven unsuccessful.⁴⁹⁸ Beyond any ambiguity, the court was prepared to assume that reverse engineers would, as a matter of course, adopt the most overall efficient reverse engineering solution. To proscribe such solution simply because it involved a greater number of intermediate copies might deny the public access to ideas and utilitarian elements that could be protected by patent law, but not by copyright.⁴⁹⁹ In contrast to *Sega*, the focus in *Sony* was not solely whether decompilation was the only way necessary in terms of technical feasibility to access the unprotected interface information but rather which one of all the available ways was the least intrusive (i.e. non-infringing) and most efficient.

2.5.3. Indispensability under the EU Software Directive

The EU Software Directive understands decompilation as of *ultima ratio* (“last resort”) to access the unprotected elements of a copyrighted computer program. The language of the Directive reads that decompilation for ends of interoperability is only permitted if “indispensable” to obtain the underlying ideas of computer code.⁵⁰⁰

Unlike U.S. jurisprudence, black-box analyses are *expressis verbis* exempted under Article 5(3) of the EU Software Directive. In lieu of permitting intermediate

⁴⁹⁴ 203 F.3d at 601.

⁴⁹⁵ *Id.*, at 600.

⁴⁹⁶ For a discussion see *infra* Part III.4.2.

⁴⁹⁷ *But see* Rothman, *supra* note 97, at 6 (stressing that the Ninth Circuit refused to draw any legal distinction between decompilation (or disassembly) and black-box methods of reverse engineering).

⁴⁹⁸ 203 F.3d at 604.

⁴⁹⁹ *Id.* at 605.

⁵⁰⁰ John T. Soma, Sharon K. Black-Keefer, Alexander R. Smith, *Minimizing Reverse Engineering: Sample Language for Dual United States and European Union Software Licenses*, 24 DENV. J. INT’L L. & POL’Y 145, 146 (1995).

copying so long as it may be justified by efficiency and legality considerations under the EU Software Directive intermediate copying as a result of decompiling computer software must be *ultima ratio* (“last resort”) in a sense of being the least non-infringing—and not expressly the most efficient—means among several technically feasible ways of uncovering unprotected elements of a program. It must though neither be the “only” way to access as demanded by the *Sega* court, nor can decompilation be justified as the most efficient way as the *Sony* court would look for, if there are still other ways technically available to access the necessary interface information.

The drafters of the EU Software Directive dealt with the same question of non-infringing alternatives as the U.S. judges of the *Sony* panel did a couple of years later. Even though both approaches refer to decompilation as kind of the “last resort” the outcomes may be different. The Ninth Circuit in *Sony* essentially decided that such alternatives should be determined by considerations of engineering efficiency and not by judicial fiat. It will be the necessity of the method of decompilation itself, not primarily the necessity of the least infringing way—of the how a defendant engineer decompiles the target program—that will render decompiling computer software permissible under U.S. copyright law.

2.6. *The Role of Interoperability in U.S. Case Law*

The second prong of the *Sega* rule requires decompilation in order to be lawful to be done for a “legitimate reason”. The question is apparent: What is a “legitimate purpose” under U.S. case law? Whereas the *Atari* court held that the fair use doctrine will be applied to permit decompilation for the purpose of understanding and copying unprotected ideas, the *Sega* court narrowed its ruling to the limited purpose of achieving compatibility only. Ultimately, the court in *Sony* at all remained silent as to determine any legitimate purpose for the applicability of the fair use defense in software decompilation cases.⁵⁰¹

2.6.1. A Contemporary Analysis of Software Interoperability

⁵⁰¹ See Prestin, *supra* note 98, at 174-5.

The EU Software Directive as of 1991 was drafted at a time when networks and the centrality of software platforms were by no means what they are today. The software industry was observed to no longer form an industry on its own, but has become foundational to a great many other industries—one may think of Web browser software to access websites and media player software for multimedia. In this context, computer software is an information platform which constitutes the foundation for the delivery and exchange of information, i.e., software and content (audio, text, video).⁵⁰²

Overbroad copyright protection will not only silence competing software programs. In fact, the danger subsists that this same copyright enables the platform owner to have extensive control on complementary expression in the form of software code, and in the form of digital content. Accordingly, the balance struck by law—as in the EU Software Directive as well as in the DMCA—ought to be about more than just software innovation as for the time being.⁵⁰³ The legal community at both sides of the Atlantic is challenged to realize that the place of software in society has changed and at the same time to keep pace with the technological advancement not only of today but also of tomorrow.

2.6.2. Shall Future Compatibility Be Protectible?

As a “highlight” of the Microsoft Judgment, the Commission’s order required Microsoft to update its interface information as it brings to market new versions of its products.⁵⁰⁴ The Commission indicated that Microsoft ultimately could upgrade its operating system so as to destroy the compatibility achieved by its rivals through their decompilation efforts.⁵⁰⁵ The Court of First Instance (CFI) affirmed the Commission Decision wherein Microsoft was ordered to share its fruits of creativeness (i.e. its interface specifications) on a long-term basis with other market players. This particular part of the Microsoft Judgment appears to contradict with the objective of Article 6(1)(c) of the EU Software Directive, where the scope of the lawful decompilation is limited to those parts of the target program which are

⁵⁰² Rotenberg, *supra* note 128, at 6 (discussing software interoperability in the European Communities in view of the freedom of expression).

⁵⁰³ *Id.* at 7.

⁵⁰⁴ Commission Decision in Microsoft, *supra* note 912, at 300.

⁵⁰⁵ *Id.* at 184-5.

necessary to be decompiled in order to achieve interoperability.⁵⁰⁶ The fact that neither the wording itself nor the legislative history provide reference to a temporal purview of the condition, invites to take a look to the pertinent U.S. copyright law, in particular to the *Atari* case.

Atari sought to copy Nintendo's entire 10NES security program to ensure compatibility with future versions of the Nintendo console. Atari feared that if it did not copy the entire 10NES program, Nintendo may update the next version of the console to monitor portions of the data stream that are currently unmonitored and in so doing to allow the console once again to distinguish between Nintendo and Atari games. Thus far the facts are congruent with Sun's line of argumentation in its complaint in the *Microsoft* case. Across the Atlantic, however, the U.S. federal court sided with Nintendo on this issue, holding that Atari may not copy portions of the program needed only for future compatibility. The court noted the importance of the balance in public policy between the dissemination of valuable ideas within an industry and the protection of a manufacturer's "lead time" in the market.⁵⁰⁷ In other words, Atari has the right to adopt Nintendo's technology, but only after Nintendo has presented it to the marketplace. If Atari could copy present code for future compatibility, it could drastically reduce the time needed to decompile and subsequently reverse engineer Nintendo's next version of the 10NES.

2.7. Two Different Standards of "Necessity"—One Unnecessary Confusion

Article 6(1)(c) of the EU Software Directive limits the decompilation exception to "the parts of the original program which are necessary to achieve interoperability."⁵⁰⁸ The Directive determines the portion allowed to copy by virtue of the overriding goal of achieving interoperability. Whether protected or unprotected elements of the target program seemingly may be copied so long as making these intermediate copies is required to reveal the interface information necessary in light of the compelling interoperability goal. It remains the question

⁵⁰⁶ EU Software Directive, *supra* note 10, art. 6(1)(c), at 45.

⁵⁰⁷ *Atari*, 975 F.2d at 843.

⁵⁰⁸ EU Software Directive, *supra* note 10, art. 6(1)(c), at 45.

whether Article 6(1)(c) is to be read as to being embedded in the broader principle of the idea/expression dichotomy or as exception thereto. The *Atari* ruling, in essence, is the U.S. answer to Europe's Article 6(1)(c).

Based on the rationale that fair use to discern a work's ideas does not justify extensive efforts to profit from replicating protected expression, intermediate copies of a computer program "must not exceed what is necessary to understand the unprotected elements of the work...[and] [a]ny reproduction of protectable expression must be strictly necessary to ascertain the bounds of protected information within the work."⁵⁰⁹

Contrary to the EU Software Directive, the court in *Atari* draws a clear distinction between disclosure of protected and unprotected elements of a copyrighted computer program. However, the ruling's significance as precedent is to be questioned in light of *Sega* and *Sony*. Some may interpret the Ninth Circuit's holding as proposition that any intermediate copying occurring during the process of decompilation, whether necessary to access the unprotected elements of the copyrighted target program or not, is a fair use. A broad interpretation like the foregoing would protect copying of a program's unprotected as well as protected elements under the guise of decompilation, even though the copying of the protected expression would not be necessary to access the unprotected elements of the target program.⁵¹⁰ In the utmost consequence, the Ninth Circuit decision in *Sony* may be read as to remove almost entirely copyright protection for object code. Software developers would be granted complete latitude in copying and manipulating the original program as long as the resulting product is transformative and thus non-infringing.⁵¹¹

While the Directive's wording is too far-reaching, the Ninth Circuit has opened the door to its courtroom for another round of litigation to determine the bounds of the fair use exception in decompilation cases.⁵¹² Given that uncertainty, object code

⁵⁰⁹ 975 F.2d 832, 843. In *Sony*, the Ninth Circuit held that Connectix disassembled parts of the PlayStation BIOS and copied the entire PlayStation BIOS multiple times. *See* 203 F.3d at 605-6. But the *Sony* court, like the court in *Sega*, neglected to address the third factor, the amount and substantiality of the portion used in relation to the copyrighted work as a whole.

⁵¹⁰ Prestin, *supra* note 98, at 169-70.

⁵¹¹ *Sony*, 203 F.3d at 604-5.

⁵¹² Due to the facts in the *Sony* case, copying of the entire PlayStation BIOS was required to complete the process of reverse engineering and to determine the protected and utilitarian portions

developers at both sides of the Atlantic may want to consider patent protection for their new programs since the scope of patent protection, unlike copyright, does protect the specific functional aspects of computer codes.⁵¹³ Alternatively, software copyright owners should consider entering into licensing agreements with their competitors instead of having them lawfully decompile a competing product.⁵¹⁴

2.8. *Decompilation of Computer Hardware*

Software needs not only to interoperate with other software, but also with hardware.⁵¹⁵ An application program usually communicates with the operating system which in turn makes the hardware perform the necessary functions. Therefore, the question is whether decompilation for the achievement of interoperable hardware may be lawful.

2.8.1. U.S. Case Law

In *Sony* court the question was examined under the forth prong of the fair use doctrine, the effect of the use upon the potential market. The court found that financial loss would predominantly accrue to Sony's hardware used to access its copyrighted works, i.e. to its PlayStation console sales and profits,⁵¹⁶ rather than to Sony's copyrighted works themselves.⁵¹⁷ Nonetheless, the court reached the conclusion that the forth factor tipped in favor of Connectix. Connectix's VGS was held not infringing Sony's copyright.⁵¹⁸ In constituting a new platform on which Sony and Sony-licensed game cartridges could be played, the VGS shows its transformative character. Connectix was a legitimate competitor to Sony in the video game console market. Consequently, Sony's attempt to monopolize this

of the program. However, the Ninth Circuit's holding may have a significant impact in future reverse engineering cases where the facts are different and thus will lead to a more contested analysis of the statutory fair use factors.

⁵¹³ See 35 U.S.C. §§ 1-376.

⁵¹⁴ Donahoe, *supra* note 65, at 111301.

⁵¹⁵ Hardware is the set of physical, tangible embodiments and structures associated with a computer system necessary to, or ancillary to the operation of the computer software—such as the keyboard, hard disk, screen, mouse and output devices. See, e.g., THOMANN & RAUBER, *supra* note 157, at 10; BAINBRIDGE, *supra* note 4, at 1.

⁵¹⁶ *Sony*, 203 F.3d at 607.

⁵¹⁷ 4 NIMMER ON COPYRIGHT, *supra* note 15, § 13.05[D][4], at 13-235.

⁵¹⁸ *Sony*, 203 F.3d at 608 n.11.

market by controlling over the market for consoles that play exclusively Sony and Sony-licensed games was rejected as to run counter the fundamental objective of copyright law that is to promote creative expression.⁵¹⁹ To conclude, the *Sony* court responded to the incipient question in affirmance: Intermediate copying as a matter of law for purpose of decompilation in order to manufacture competing hardware was held permissible under the fair use doctrine. Does European copyright law provide for a statutory pendant to the U.S. approach adopted in *Sony*?

2.8.2. The European Approach

The EU Software Directive does not unambiguously provide an answer to the question. However, from a semantic interpretation of the Directive may be inferred an affirmative position for allowing decompilation also for the achievement of interoperable hardware. Implicitly the drafters of the Directive distinguished between the “computer program” and “computer system.” The term “computer program” is defined in Recital 7 as any program in any form.⁵²⁰ In Recital 10 the function of a computer program is explained as “to work together with other components of a computer system with users.”⁵²¹ Again, in Recital 23 the Directive refers to the terms’ different scopes by defining decompilation for the purpose of making “it possible to connect all components of a *computer system*, including those of different manufacturers, so that they can work together.”⁵²² Thus, the term “computer system” as used in Recitals 10 and 23 appears in a broader meaning including both software and hardware components as being subject to decompilation under copyright law.

⁵¹⁹ 203 F.3d at 607-8 (recognizing some economic loss by Sony as a result of competition in the video game console market which could not be sought to be prevented by finding no fair use because copyright law may not be misused by Sony to confer a monopoly).

⁵²⁰ EU Software Directive, *supra* note 27, recital 7, at 42 (including “programs in any form, including those which are incorporated into hardware; whereas this term also includes preparatory design work leading to the development of a computer program provided that the nature of the preparatory work is such that a computer program can result from it at a later stage”).

⁵²¹ *Id.*, recital 10, at 43, reads in full text:

“Whereas the function of a computer program is to communicate and work together with other components of a computer system and with users and, for this purpose, a logical and, where appropriate, physical interconnection and interaction is required to permit all elements of software and hardware to work with other software and hardware and with users in all the ways in which they are intended to function[.]”

⁵²² *Id.*, recital 23, at 43 (emphasis added).

At the same time, an opposing standpoint emerged in Europe. First, because any hardware's role during the legislative drafting was mentioned⁵²³ and so does neither the clear wording of Article 6 of the EU Software Directive itself, arguably the boundaries for gap filling *per analogiam* are to be of a limited nature.⁵²⁴ Second, hardware is subject to a different legal protection regime and can in most instances be decompiled in order to analyze its construction.⁵²⁵ Third, the comments and recitals above-mentioned relate to the development of new interconnecting software interfacing between existing software and a piece of hardware, and not to the creation of new hardware. In spite of the technical interrelation between software and hardware (i.e., software programs the hardware to be hardware)⁵²⁶ there is a difference in the process of obtaining interface information for the purpose of interoperability. Whereas decompiling computer code every time⁵²⁷ implies a reproduction of the part of the program expressing the interface specifications as there is no physical manifestation of the interface which can be otherwise observed, there is always a physical link between the hardware device and the hardware device storing and executing the program to be interfaced with. In the case of hardware devices, the functions of the interface can, thus, be observed by monitoring the signals communicated across the physical interface when executing the program interfacing with the hardware.⁵²⁸ For example, it is not necessary to decompile the printer device software to obtain the printer interface protocols for the design of a new printer which will emulate existing printers driven from the printer driver software since the printer interface can be observed in operation by recording the signals that pass across the physical interface between

⁵²³ HABERSTUMPF in LEHMANN, *supra* note 2, at 162 (noting that the proposal to permit decompilation also for the purpose of interoperability with hardware was refused by both the European Parliament and the Council).

⁵²⁴ WALTER, *supra* note 38, at 230.

⁵²⁵ CZARNOTA & HART, *supra* note 52, at 85.

⁵²⁶ Cf. Joe Linhoff, *Video Games and Reverse Engineering: Before and After the Digital Millennium Copyright Act*, 3 J. TELECOMM. & HIGH TECH. L. 209, 219-20 (2004) (advocating an approach based on analogy since the distinction between hardware and software is blurred, and as copyright reaches to protect software, it also protects hardware); see also ULLRICH, *supra* note 76, at 80 (exemplifying operating system programs as within the scope of Article 6 while indicating the difficulty to separate the levels of hardware and software).

⁵²⁷ In certain circumstances repeatedly acts of decompiling software may be necessary and permissible. LEHMANN, *supra* note 2, at 25 n.98.

⁵²⁸ CZARNOTA & HART, *supra* note 52, at 85; *contra* MARLY, *supra* note 111, at 323.

the printer and the processor port.⁵²⁹ Additionally, the actual electronic functions performed by the printer circuitry can be ascertained from an analysis of the circuit diagram of the printer electronics.⁵³⁰

V. The Reverse Engineering Exception Under The Digital Millennium Copyright Act

1. A Piece Of Legislation to Keep Pace With the Technological Advancement

1.1. WIPO's Influence on the Battle Between Hollywood and Silicon Valley

The inherent conflict between the exclusive rights of software copyright owners and the fair use rights of the general public subsists at both ends of the Atlantic. So, like in Europe as of almost a decade earlier, there came up two fractions dominating the decompilation battle in the United States. “Hollywood”, as one interested group, sought strong protections for the owners of original works and the technology used to protect such works, while “Silicon Valley”, as the other interested group, opposed broad protections that would impede on the ability to engage in decompiling computer programs, computer security testing, and encryption research.

The desire for an extra level of protection for copyright owners inspired the anti-circumvention provisions included in Article 11 of the WCT.⁵³¹ Article 11 requires each contraction nation to address the circumvention of effective technological measures that are used by copyright owners to protect their exclusive rights. The United States signed the WCT in 1997. In response to comply with the WIPO

⁵²⁹ This is rather unclear as regards the interconnection of software and hardware, *see* LEHMANN, *supra* note 2, at 24 n.95.

⁵³⁰ CZARNOTA & HART, *supra* note 52, at 85.

⁵³¹ *See* WCT, *supra* note 191, 36 I.L.M. at 71, requires member states to “[p]rovide adequate legal protection and effective legal remedies against the circumvention of effective technological measures that are used by authors in connection with their exercise of their rights under this Treaty or the Berne Convention and that restrict acts...which are not authorized by the authors concerned or permitted by law.”

mandate and to make necessary changes in U.S. copyright law, the United States enacted the Digital Millennium Copyright Act in 1998.⁵³² The purpose of the DMCA expressed in its introduction was a reiteration, almost verbatim, of the WCT.

1.2. *The Anti-Circumvention Provisions under Section 1201(a) of the DMCA*

The DMCA was primarily drafted to protect technology used by copyright owners to protect their digital works from being outrun by technology that would allow others to circumvent their protection.⁵³³ Section 1201(a)(1)(A) of Title 17 of the Copyright Act as the general rule provides that “[n]o person shall circumvent a technological measure that effectively controls access to a work protected.”⁵³⁴ In addition, the DMCA prohibits trafficking in technologies that are designed to circumvent technological safeguards.⁵³⁵ The key to liability under section 1201 is the circumvention or the marketing of a prohibited device. Whether the circumventing actions constitute infringement of copyright is not relevant to the applicability of section 1201(f)(1). Violations of the anti-circumvention provisions are not copyright infringement yet independent violations separately actionable under section 1201.⁵³⁶

In the software reverse engineering context, the DMCA is implicated at three stages. The first time is during the decompilation phase when the reverse engineer

⁵³² Digital Millennium Copyright Act, Pub. L. No. 105-304, 112 Stat. 2860 (1998) [hereinafter DMCA].

⁵³³ David N. Pruitt, *Beyond Fair Use: The Right to Contract Around Copyright Protection of Reverse Engineering in the Software Industry*, 6 CHI.-KENT J. INTELL. PROP. 66 (2006) (characterizing the DMCA as “burglar’s tools” statute applying to those who were using circumvention techniques to illegally obtain copyrighted digital materials, as opposed to those who entered into licensing agreements to obtain the materials).

⁵³⁴ 17 U.S.C. § 1201(a)(1)(A) (2000).

⁵³⁵ 17 U.S.C. § 1201(a)(2) (2000) in full text:

“No person shall manufacture, import, offer to the public, provide or otherwise traffic in any technology, product, service, device, component, or part thereof, that—

(A) is primarily designed or produced for the purpose of circumventing a technological measure that effectively controls access to a work protected under this title;

(B) has only limited commercially significant purpose or use other than to circumvent a technological measure that effectively controls access to a work protected under this title, or

(C) is marketed by that person or another acting in concert with that person with that person’s knowledge for use in circumventing a technological measure that effectively controls access to a work protected under this title.”

⁵³⁶ In literature there is referred to a distinction between traditional copyright and the “paracopyright” status of the DMCA.

analyzes how the existing computer code (i.e., the digital lock) works and determines what measures are needed to unlock that program and what are the requirements for achieving interoperability. The second time is when the reverse engineer develops his own digital key that works with the existing program. The third time is when the reverse engineer finally embeds his own digital key into his new product, so that the new product enables unlocking the underlying program. For the purpose of this paper the emphasis lies on the first phase, the decompilation of computer programs.

1.3. *The Reverse Engineering Exception*

1.3.1. Section 1201(f)(1) of the DMCA

The prohibition against circumvention of protective measures is not absolute as to the intermediate copyist.⁵³⁷ The “safe harbor” provision of section 1201(f)(1) prescribes that

[n]otwithstanding the [general rule] of subsection (a)(1)(A), a person who has lawfully obtained the right to use a copy of a computer program may circumvent a technological measure that effectively controls access to a particular portion of that program for the sole purpose of identifying and analyzing those elements of the program that are necessary to achieve interoperability of an independently created computer program with other programs, and that have not previously been readily available to the person engaging in the circumvention, to the extent any such acts of identification and analysis do not constitute infringement under this title.

The exemption covers the circumvention of both access and copy controls and permits the development of devices or technologies to circumvent access and copy controls if necessary to accomplish interoperability.⁵³⁸ The text of section 1201(f)(1) makes explicit that not all intermediate copyists fall within the scope of the provision. The provision narrowly limits the range of circumstances in which circumvention was accomplished by restricting the act of reverse engineering.

⁵³⁷ See generally *Universal City Studios, Inc. v. Reimerdes*, 111 F.Supp. 2d 294, 315-6 (S.D.N.Y. 2000) [hereinafter *Reimerdes*]; *Universal City Studios, Inc. v. Corley*, 273 F.3d 429 (2d Cir. 2001) [hereinafter *Corley*].

⁵³⁸ MARSHALL A. LEAFFER, UNDERSTANDING COPYRIGHT LAW (2005) 400.

Outside of those narrow contexts reverse engineering is not a protected right.⁵³⁹ But even within the narrowly drawn limits of interoperability throughout the reverse engineering exemption of section 1201(f), there are ambiguities that U.S. courts must resolve to apply the reverse engineering exemption.

1.3.2. The Scope—A Bill of Conditions

First, the person performing the act of circumvention must possess a lawfully obtained right to use a copy of a computer program. Second, a reverse engineer may develop a compatible digital key and employ it in order to analyze the interoperability requirements. Interoperability is defined as “the ability of computer programs to exchange information” so that the information exchanged can be used by such programs.⁵⁴⁰ Unlike the yet unclear objection of Article 6 of the EU Software Directive, the text specifies only program-to-program interoperability, while digital locks and keys in hardware do not come under this exception.⁵⁴¹ Section 1201(f)(1) clearly delineates the acts of circumvention itself from the end-results accomplished through acts of circumvention. The interoperable program as result of the circumvention must be an “independently created” work. According to general principles of copyright law, the new program must satisfy the threshold of the necessary modicum of original expression to enjoy protection under the U.S. Copyright Act. In other words, the new program must not infringe on the copyrightable aspects of the original.⁵⁴²

Third, section 1201(f)(1) limits the scope to “a particular portion” of the program, that is the portion that contains the “elements of the program that are necessary to achieve interoperability...and that have not previously been readily available.”⁵⁴³ It is practically unlikely for a reverse engineer to know ahead of time which particular portion of a program to examine. Even after decompiling an entire program, it may not be obvious which elements of the program are necessary to achieve interoperability. In such an event, the reverse engineer must take into

⁵³⁹ See *Edelman v. N2H2, Inc.*, 263 F.Supp.2d 137 (D. Mass. 2003) (no right or privilege to reverse engineer Internet filtering software).

⁵⁴⁰ See 17 U.S.C. § 1201(f)(4) (2000).

⁵⁴¹ Meninsky, *supra* note 100, at 612.

⁵⁴² See S. REP. No. 105-190, at 32 (1998) (“The resulting program must be a new and original work, in that it may not infringe the original computer program.”).

⁵⁴³ See 17 U.S.C. § 1201(f)(1) (2000).

account material that did not survive translation into object code in the first place, such as source code commentary and program specifications, to determine compatibility requirements.⁵⁴⁴ Even if interface specifications are published and thereby making those elements necessary to achieve interoperability “readily available”, the software producer may not have obviated the need to decompile the target program. A static examination of certain elements of a program is no substitute for an analysis of the entire program as it is operating.⁵⁴⁵ After all, the general idea behind section 1201(f) is that the reverse engineering exception is available only if a program’s interface specifications are not accessible by other means.⁵⁴⁶

Lastly, circumvention of a technical measure is allowed only to the extent the intermediate copyist’s actions do not meet the copyright infringement threshold. The words “to the extent” appear to invite courts to undergo a fact-intensive analysis of the potentially infringing intermediate copies made by the programmer in the course of reverse engineering.

Because of the akin wording used in the EU Software Directive of 1991, these ambiguities naturally are reminiscent to those we encountered at European level.⁵⁴⁷ Given the language used in section 1201(f)(1) and the incorporation of eight fair use exemptions in the DMCA, one of them which is the reverse engineering exemption, the DMCA appears as manifest of protecting fair access for the reverse engineering community. At least on its face, by adopting the DMCA to protect at least a limited right to reverse engineer, including the right to decompile, the United States seem to be in line with Europe’s policy favoring open systems. With a closer look to the underlying policies in either jurisdiction that assertion is not free of objections.

1.4. *Sony v. DMCA*

⁵⁴⁴ Johnson-Laird, *supra* note 7, at 899.

⁵⁴⁵ Donna L. Lee, *Reverse Engineering of Computer Programs under the DMCA: Recognizing a “Fair Access” Defense*, 10 MARQ. INTELL. PROP. L. REV. 537, 555.

⁵⁴⁶ LEAFFER, *supra* note 538, at 401.

⁵⁴⁷ See EU Software Copyright Directive, *supra* note 10, art. 6, at 45; see also Markus Fallenbock, *On The Technical Protection Of Copyright: The Digital Millennium Copyright Act, the European Community Copyright Directive and Their Anticircumvention Provisions*, 7 INT’L J. COMM. L. & POL’Y 4, 30 (2002).

Basically, the *Sony* court seems to be saying that we should not be concerned with the process but with the end-result. In other words, U.S. laws should not restrict intermediate steps or products. Instead, U.S. laws should prohibit only finished products that infringe exclusive copyrights, or the harmful use of non-infringing products (e.g., using a personal computer to upload infringing material onto the Internet). Nevertheless, Congress did not follow this approach when they drafted the DMCA. Congress seemingly has codified the *Sega* ruling wherein the court clearly separated the question of copyright infringement from the question of whether the fair use defense applies to decompilation.

In *Reimerdes*, strict application of the DMCA's anti-circumvention provisions would inhibit the development of a Linux-compatible DVD drive.⁵⁴⁸ This is because the DMCA prohibits technologies which circumvent measures designed to prevent unauthorized access to or reproduction of copyrighted works, even if the technologies have a substantial non-infringing use. A Linux compatible DVD drive could be used, for example, to play a lawfully purchased DVD on a computer with a Linux operating system. By inhibiting the development of Linux compatible DVD drives, the DMCA hinders the ability of Linux to compete with Microsoft Windows. Linux can emerge as a significant competitor to Windows only if it offers all the functionality of Windows. Certainly, Congress did not intend this result at the time enacting the DMCA. But such unintended consequences occur when the law focuses on intermediate steps—the technological act of circumvention in and of itself⁵⁴⁹—rather than the end result.⁵⁵⁰

⁵⁴⁸ See, e.g., *Reimerdes*, 111 F.Supp.2d 294 (the United States Court of Appeals for the Second Circuit holding that if a software user successfully obtained access to source code by circumventing a technological “lock” or barrier without authorization from the copyright holder of the computer program, the circumvention would not be excused by the doctrine of reverse engineering under the DMCA unless the defendant could persuade the court that her purpose for circumvention was to determine how the two programs interoperate).

⁵⁴⁹ Cf. 17. U.S.C. § 1201(f)(1) reading in its pertinent part as: “[A] person...may circumvent a technological measure...to the extent *any such acts of identification and analysis* do not constitute infringement under this title.” (emphasis added)

⁵⁵⁰ But see Lee, *supra* note 545, at 559-60 (stating a deviation from the *Sega* test by Congress masking out the intermediate copying stage and instead directly focusing on the programmer's interoperable program; to the view of the author, Congress collapsed the two elements of the *Sega* rule into a single threshold question for a court to ask: Did the defendant reverse engineer the plaintiff's program with the purpose of creating a competing, interoperable program?).

2. A Fair Re-Balance under the DMCA?

2.1. *The Pre-DMCA Balance*

Prior to the enactment of the DMCA, reverse engineering exposed the engineer to two potential claims of copyright infringement. The first occurred during the analysis phase, to wit, making intermediate copies in the course of decompiling the target program in violation of the copyright owner's exclusive reproduction right. The second was in creating the new computer program.⁵⁵¹ Under copyright law, intermediate copies in the process of reverse engineering were generally upheld as fair use when decompilation was necessary to obtain knowledge of unprotected aspects underlying the target program,⁵⁵² the reverse engineer had a legitimate interest in gaining such access⁵⁵³ and a new non-infringing product was created. The rationale was found in public interests associated with reverse engineering generally. The net benefit to the public lies in promoting dissemination of unprotected information, and thereby furthering the creation of new, innovative products.⁵⁵⁴

2.2. *The Post-DMCA Balance*

Three circumstances call for discussing whether the DMCA entails a re-balance of the current copyright regime in the United States: the narrow purpose of interoperability, the broad interpretation of circumventing acts, and its effect on legalizing copyright misuse. The computing industry was afraid that Congress by focusing on the technological act of circumvention in and of itself, as opposed to copyright infringement, the DMCA will create a significant diminution of fair use.

⁵⁵¹ Meninsky, *supra* note 100, at 594.

⁵⁵² See, e.g., *Atari*, 975 F.2d at 843 (permitting fair use protection for reverse engineering when necessary to discover technology that could not be otherwise discovered).

⁵⁵³ See, e.g., *Sega*, 977 F.2d at 1518 (holding fair use defense to apply "because...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...When there is a good reason for studying or examining the unprotected aspects of a copyrighted computer program, [decompilation] for purposes of such study or examination constitutes fair use.").

⁵⁵⁴ See *Bateman v. Mnemonics, Inc.*, 79 F.3d at 1532, 1540 (11th Cir. 1996) (confirming the *Sega* opinion in view of the principal purpose of copyright stated to be "the advancement of science and the arts").

If the DMCA does not use copyright as the decisive criterion for violation of the copyright work, then fair use cannot be a limitation on liability.⁵⁵⁵ Did Congress enact the DMCA in awareness of the given outgrowth? Are these ramifications, even if unintended, justifiable from a copyright policy standpoint?

2.2.1. Interoperability

Legislative drafting of the DMCA took place with *Sega* in mind while proclaiming that the “objective is to ensure that the effect of current case law interpreting the Copyright Act is not changed by enactment of this legislation.”⁵⁵⁶ However, the narrowly defined focus of the DMCA does not go as far as prior case law in allowing for reverse engineering in several respects.⁵⁵⁷ The *Sega* court held that reverse engineering for “a legitimate reason” constituted fair use.⁵⁵⁸ Whereas *Sega* could be extended for general non-exploitative purposes,⁵⁵⁹ six years later, the DMCA limited this broad approach to the sole purpose of achieving interoperability. Succinctly, the fair access to copyrighted computer code via reverse engineering is reduced to accomplish interoperability.

Assuming that a computer programmer engages in acts of decompilation of a rival’s program for the non-exploitative purpose of extracting public domain algorithms from it, decompiling the competitor’s program would qualify under the fair use doctrine but would fall outside the safe harbor of the DMCA because the rival’s program would not be decompiled for the purpose of achieving interoperability.⁵⁶⁰ As the legislation currently stands, the reverse engineer could not raise the fair use defense as a defense against DMCA liability for unauthorized circumvention or trafficking in a circumvention device,⁵⁶¹ even though the

⁵⁵⁵ *Copyright Treaties Implementation Act: Hearings on H.R. 2280, 2281 Before the Subcomm. on Courts and Intellectual Property of the H. Comm. on the Judiciary, 105th Cong. (1997)* (statement of Edward J. Black, President, Computer & Communications Industry Assn.).

⁵⁵⁶ See S. REP. No. 105-190, at 13, 32 (1998); H.R. REP. No. 105-796, at 42 (1998) (Conf. Rep.).

⁵⁵⁷ 4 NIMMER ON COPYRIGHT, *supra* note 15, § 13.05[D][4], at 13-233-4.

⁵⁵⁸ *Sega*, 977 F.2d at 1518.

⁵⁵⁹ Albeit all of the cases that have dealt with reverse engineering thus far have involved interoperability, a case could come up when the court would need to articulate a more general, fair use defense for reverse engineering undertaken for a legitimate reason other than interoperability.

⁵⁶⁰ 4 NIMMER ON COPYRIGHT, *supra* note 15, § 13.05[D][4], at 13-234 n.513.14.

⁵⁶¹ See *Reimerdes*, 111 F.Supp. 2d at 322 (holding the fair use defense to fail because fair use is a defense to copyright infringement and defendants were not sued for copyright infringement but for anti-circumvention trafficking; Congress provided no fair use defense to this action), *see also*

resulting use of the copyrighted program to be decompiled could be excused as a fair use.⁵⁶²

2.2.2. Focusing On Circumventing Acts

Most DMCA cases to date have dealt with manufacturers who put digital locks on their devices, either through software authorization sequences or encryption keys, to prevent competitors from gaining access to their devices.⁵⁶³ The purpose of the digital locks in these cases is to protect digital content from piracy. On the other hand, the reverse engineer who seeks to achieve interoperability is creating his own competing content in form of a new computer program that is compatible with the existing program, rather than pirating existing content. Circumvention of an access or copy control is usually accomplished through reverse engineering of the pertinent control device. The DMCA, however, does not make this distinction. The DMCA misses to focus on the engineer's intent underlying the interoperability purpose and instead applies the same measurement of interoperability for both ends.

2.2.3. Legalized Copyright Misuse under the Auspices of the DMCA?

In analogy to patent misuse,⁵⁶⁴ the copyright misuse doctrine⁵⁶⁵ may prohibit a copyright owner from extending the limited monopoly beyond the limited

Corley, 273 F.3d at 443 (2d Cir. 2001) (holding fair use was not a defense to the DMCA because the DMCA was not a copyright provision and the DMCA does not deal with the use of copyrighted material after the circumvention occurs).

⁵⁶² *Corley* challenged the DMCA for restricting his right of fair use under the Intellectual Property Clause and the First Amendment of the U.S. Constitution. The court found no Constitutional collisions since fair use was not constitutionally required, nor had fair use ever been "held to be a guarantee of access of copyrighted material in order to copy it by the fair user's preferred technique or in the format of the original." *Corley*, 273 F.3d at 436, 458-9 (2d Cir. 2001). See also Meninsky, *supra* note 100, at 614-6, 621-2 (discussing Constitutional aspects of the DMCA).

⁵⁶³ See *Atari*, 975 F.2d 832 (Fed. Cir. 1992); *Sega*, 977 F.2d 1510 (9th Cir. 1992) (dealing with video game console manufacturers who tried to lock out competing game cartridge developers from using their consoles).

⁵⁶⁴ For a discussion on copyright misuse see *infra* Part VII.1.2. See also *Motion Picture Patents Co. v. Universal Film Manufacturing Co.*, 243 U.S. 502 (1917); *Morton Salt Co. v. G.S. Suppiger Co.*, 314 U.S. 488 (1942).

⁵⁶⁵ Currently, a split of authority exists among the federal circuit courts that have addressed the doctrine of misuse as applied to copyright law. For example, *DSC Communications Corp. v. DGI Technologies*, 81 F.3d 597, 601 (5th Cir. 1996) has explicitly considered the copyright misuse doctrine. On the other hand, in *Saturday Evening Post Co. v. Rumbleseat Press*, 816 F.2d 1191 (7th Cir. 1987) the Seventh Circuit has rejected the copyright misuse doctrine.

monopoly granted to him by the copyright laws.⁵⁶⁶ The intellectual property misuse defense in general is rooted in the traditional “unclean hands” doctrine in equity law. Under the “unclean hands” doctrine, courts will deny an otherwise meritorious claim when the plaintiff has acted so improperly that the need to punish the plaintiff’s wrongful behavior outweighs the need to punish the defendant’s unlawful conduct.⁵⁶⁷ In “paracopyright” cases under the DMCA, due to the same reasons why courts do not apply a fair use analysis, U.S. courts reluctantly, if at all, employ the copyright misuse defense—causing in effect the same baleful results when denying the fair use defense to be invoked.⁵⁶⁸

Since a digital lock is a functional, non-protectible element to protect a program’s underlying interface information, enforcing that lock will extend a copyright holder’s exclusive rights beyond the valid limits of his copyright in two respects. First, the DMCA’s general anti-circumvention provisions are of unlimited duration. What was once a copyright of limited duration under the Copyright Act⁵⁶⁹ now has taken on the form of a patent with unlimited duration under the DMCA.⁵⁷⁰ Second, the anti-circumvention provisions of the DMCA would warrant copyright protection of otherwise uncopyrightable aspects of computer programs. It was the latter what proved the defendant’s undoing in *Lexmark International Inc. v. Static Control Components, Inc.*⁵⁷¹ The defendant reverse engineered the Lexmark printer

⁵⁶⁶ See generally *Lasercomb America, Inc. v. Reynolds*, 911 F.2d 970 (4th Cir. 1990); *A&M Records, Inc. v. Napster, Inc.*, 239 F.3d 1004 (9th Cir. 2001).

⁵⁶⁷ *Broadcast Music, Inc. v. Hearst/ABC Viacom Entertainment Services*, 746 F.Supp. 320, 329-30 (S.D.N.Y. 1990) (citing Nimmer on Copyright).

⁵⁶⁸ Cf. Jacqueline Lipton, *The Law of Unintended Consequences: The Digital Millennium Copyright Act and Interoperability*, 62 WASH. & LEE L. REV. 487, 532, 540 (stating that, as yet, the legislation or jurisprudence likewise lacks guidance as to whether the copyright misuse defense might validly be raised in a DMCA infringement action).

⁵⁶⁹ Cf. U.S. CONST. art. I, § 8, cl. 8 (granting Congress the power “[t]o promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries”) [hereinafter Intellectual Property Clause].

⁵⁷⁰ The legislative history indicates that Congress expressly enacted the DMCA based on its authority under the Commerce Clause rather than its authority under the Intellectual Property Clause. See H.R. REP. NO. 105-551, at Title I § 107(d) (1998). If the DMCA is not fundamentally inconsistent with the Intellectual Property Clause and is otherwise within Congress’ authority under the Commerce Clause, then the DMCA is not an unconstitutional exercise of congressional power. Two courts have questioned whether the perpetual anti-circumvention provisions of the DMCA under the Commerce Clause meet the limited times requirement of the Intellectual Property Clause. *Corley*, 273 F.3d at 445 (2d Cir. 2001), *United States v. Elcom Ltd.*, 203 F.Supp. 2d 1111, 1141 n.8 (N.D. Cal. 2002); see also Stephen E. Blythe, *The U.S. Digital Millennium Copyright Act And The E.U. Copyright Directive: Comparative Impact On Fair Use Rights*, 8 TUL. J. TECH. & INTELL. PROP. 111, 122 (2006).

⁵⁷¹ 387 F.3d 522 (6th Cir. 2004) [hereinafter *Lexmark*].

and cartridge software. The cartridge chip contained a 37-byte program which informed the printer when the ink was low in the cartridge.⁵⁷² The defendant thought that the digital key for Lexmark's digital lock consisted of the printer chip receiving the proper "ink full" message from a replacement cartridge and the 37 bytes were the authentication sequence. The defendant remanufactured cartridges by refilling them with ink and inserting its own chip that generated Lexmark's "in full" message. In reality, however, only 7 bytes in specific memory locations on the Lexmark cartridge chip served as the authentication sequence.⁵⁷³ Thus, the defendant took more than what was essential to achieve interoperability. Lexmark succeeded in narrowing the focus of the court to the solely on its program, instead of viewing the program as merely an embedded component in controlling the operation of a printer. The defendant opined that the primary product was the newly refilled ink cartridge in which it ancillary had incorporated a digital key. The cartridge and ink were merely staple articles protected by neither patent nor copyright. As a result, Lexmark could not only extend its patent and copyright monopolies to replacement cartridges, but to the sale of replacement ink as well.

In conclusion, by focusing on the reverse engineered digital lock (i.e. the program) itself, and not on what the lock seeks to protect, the DMCA is thus implicitly legalizing copyright misuse by granting protection of otherwise non-protectible interface information. As a consequence, the DMCA is a powerful tool to "lock-out" competition as it bars innovators from creating new and improved products that interoperate with existing products. Particularly, small secondcomers will be denied entering the market because they lack of resources for developing and marketing "the wheel" anew. For all these aforementioned reasons, U.S. courts should be alert of plaintiffs with "unclean hands" claiming protection under the DMCA and narrow the reach of the DMCA's anti-circumvention provisions in those situations since the legislator did not provide for any mechanism in the DMCA similar to the copyright misuse doctrine under copyright law.

The legislative history reveals that Congress wanted to ensure that section 1201 would not be asserted against persons who engaged in reverse engineering of

⁵⁷² *Lexmark*, 387 F.3d at 529.

⁵⁷³ 387 F.3d at 530-1.

copyrighted works.⁵⁷⁴ In the belief to preserve the development of new technology, Congress inserted the reverse engineering exemptions so that the DMCA would not “constitute a serious impediment to the development and production of competitive goods and services.”⁵⁷⁵ Unfortunately, the unintended consequences prove otherwise. In resolving this quandary, courts might want to broadly interpret Congress’ intent according to which the fair use doctrine should not be “frozen” in the Copyright Act, but should remain in the hands of the judiciary, so that it could more easily respond to technological changes.⁵⁷⁶ At the end of the day, it therefore remains to be seen whether case law under the fair use doctrine will diverge from interpretation of the DMCA, or whether the DMCA will itself be construed expansively.

3. The EU Software Directive Meets Section 1201(1)(f) of the DMCA: Same Wording, but Different Policies

The anti-circumvention provision is said to have the greatest impact on the fair use doctrine.⁵⁷⁷ Therefore, many critics have charged the narrow protection adopted by the DMCA as to overall threaten the balance between a copyright holder’s rights, fair use, and competition by severely restricting the software developers’ ability to legally reverse engineer.⁵⁷⁸ For these reasons, U.S. legislators may be said to have taken a step backwards from Europe’s open system model. The balance between the exclusive rights of U.S. software copyright owners and the right for the general

⁵⁷⁴ Pruitt, *supra* note 533, at 66.

⁵⁷⁵ See 144 CONG. REC. H7079 (daily ed. Aug. 4, 1998).

⁵⁷⁶ See H.R. REP. NO. 94-1476, at 67 (1976) (“The bill endorses the purpose and general scope of the judicial doctrine of fair use, but there is no disposition to freeze the doctrine in the statute, especially during a period of rapid technological change. [T]he courts must be free to adapt the doctrine to particular situations on a case-by-case basis.”).

⁵⁷⁷ See generally Stephen E. Blythe, *The U.S. Digital Millennium Copyright Act And The E.U. Copyright Directive: Comparative Impact On Fair Use Rights*, 8 TUL. J. TECH. & INTELL. PROP. 111, 117 (2006).

⁵⁷⁸ Jeffrey D. Sullivan & Thomas M. Morrow, *Practicing Reverse Engineering In An Era Of Growing Constraints Under The Digital Millennium Copyright Act And Other Provisions*, 14 ALB. L. J. SCI. & TECH. 1, 6 n.17 (2003) (listing common situations in which reverse engineering of programs of competitors would previously have appeared commercially legitimate, such as investigating another’s system to detect patent or copyright infringement, learning the concept behind the system, repairing the product, providing a service usable with the system, improving the product, and that now must eliminate reverse engineering practices because of falling within the ambit of the DMCA); see also Linhoff, *supra* note 369, at 229.

reverse engineering public in a free society of unfettered access to computer source codes has tilted toward more property rights for owners of copyrighted software by way of limiting fair use rights for reverse engineers.

Although the decompilation exemption is likewise restricted to the sole purpose of achieving interoperability, the European legislator has taken a step forwards by adopting Article 6 of the EU Software Directive because prior to Article 6 the question of software decompilation was not addressed at all by EU copyright law. In spite of some non-harmonious inaccuracies, this piece of legislation was the ambitious beginning of making source codes accessible to the public on Europe's way to open systems.

After all, the difference in terms of U.S. and European copyright policy is that the legislators of section 1201(f) of the DMCA have instrumentalized the interoperability criterion in a way to restrict the broad and flexible mechanism of the fair use doctrine in software reverse engineering (including decompilation) cases, whereas the legislators of Article 6 of the EU Software Directive have made use of the interoperability with the intention of fostering software decompilation with view to the international computer software market on the basis of an open system model.

VI. Another Strategy in the Decompilation Battle: Shrink-wrap and Click-on Agreements

In both Europe and the United States, creative efforts generally are protected by intellectual property law, trade secret law and contract law.⁵⁷⁹ In terms of software protection this paper so far has dealt with Article 6 of the EU Software Directive as landmark legislation harmonizing decompilation within Europe and, in the United States, with federal legislation protecting reverse engineering by section 107 of the Copyright Act and section 1201(f) of the DMCA. An alternative to "lock out" decompilers and reverse engineers provides contract law. Why would software

⁵⁷⁹ Soma, Black-Keefer & Smith, *supra* note 500, at 147.

providers want to include license terms that are more restrictive than the rights these pieces of legislation would otherwise confer?

Source code is the fruit of usually substantial investment. Software providers thus have an interest in utmost protection of source code as well as the trade secrets contained therein. To that effect software developers resort to state contract law to control, *inter alia*, decompilation of their computer program code. Typically software owners do so in so-called “shrink-wrap”⁵⁸⁰ and “click-on”⁵⁸¹ licenses to which they demand adhesion as a condition to licensing their software package.

Most software bought in the open market today subjects the (end) user to a break-the-seal contract, a shrink-wrap contract.⁵⁸² Beneath the clear outer wrapping is a visible license agreement that states that by opening the seal the user agrees not to copy, decompile or modify the program. Such shrink-wrap agreement is also frequently present on an envelope containing the software inside the sealed outer box. It states that if the buyer does not consent to this licensing agreement, he or she should return the unopened software to the place of purchase. Click-on software license are the more contemporary version of shrink-wrap agreements. In lieu of agreeing to the contract by opening the packaging, the user agrees to the click-on contract by clicking an “I accept” box while installing the software on a computer.⁵⁸³

1. Contractual Restrictions under the EU Software Directive

1.1. Article 9(1) of the EU Software Directive

⁵⁸⁰ Also termed, for example, “box-top” license or “tear-me-open” license.

⁵⁸¹ Also known in literature under synonyms such as “click-through” or “click-wrap” license agreements or “point-and-click” agreements.

⁵⁸² BLACK’S LAW DICTIONARY 639 (8th ed. 2004) (defining a shrink-wrap license as “[a] license printed on the outside of a software package to advise the buyer that by opening the package, the buyer becomes legally bound to abide by the terms of the license. Shrink-wrap licenses usu[ally] seek to (1) prohibit users from making unauthorized copies of the software, (2) prohibit modifications to the software, (3) limit use of the software to one computer, (4) limit the manufacturer’s liability, and (5) disclaim warranties.”).

⁵⁸³ BLACK’S LAW DICTIONARY 1195 (8th ed. 2004) (defining a click-wrap license as “[a]n electronic version of a shrink-wrap license in which a computer user agrees to the terms of an electronically displayed agreement by pointing the cursor to a particular location on the screen and then clicking.”).

As general rule, the EU Software Directive leaves it with the law of contract in a Member State to permit or prohibit acts of reverse engineering. Only in respect of decompilation (Article 6), making of a back-up copy (Article 5(2)) and black-box analysis (Article 5(3)), the Directive regulates possible conflicts between the Directive and national contract laws.⁵⁸⁴

According to the second sentence of Article 9(1) of the EU Software Directive⁵⁸⁵ the decompilation exception constitutes *ius cogens*; i.e., as a mandatory provision, any contractual provisions contrary to Article 6 and to the subparagraphs (2) and (3) of Article 5⁵⁸⁶ will be considered null and void. Hence, the statutory exemption, such as permissible acts of decompiling, cannot be overridden by contractual provisions. For example, the German Copyright Act, in section 69(g)(2) follows suit and explicitly states that contracting out of decompilation is not permitted.⁵⁸⁷ However, the second sentence of Article 9(1) does not guarantee communication flow of information by itself, but rather disclosure, if any at all, and, if so, the amount of information to be disclosed solely depends on the software owner's will. First, the software developer must license his computer software, then, and only then, the right of a lawful licensee to engage in certain techniques of analyzing computer software cannot be abridged by the software owner.⁵⁸⁸ Conversely, if there is no license agreement, there will be no minimum guarantee on decompilation or Article 5(3) techniques. Therefore, Article 9(1)'s purpose to warrant a minimum right to decompile computer programs is of relative significance in practice.

⁵⁸⁴ CZARNOTA & HART, *supra* note 52, at 96.

⁵⁸⁵ EU Software Directive, *supra* note 10, art. 9(1), at 45, says:

“Continues application of other legal provisions.

1. The provisions of this Directive shall be without prejudice to any other legal provisions such as those concerning patent rights, trade-marks, unfair competition, trade secrets, protection of semiconductor products or the law of contract. Any contractual provisions contrary to Article 6 or to the exceptions provided for in Article 5(2) and (3) shall be null and void.”

⁵⁸⁶ If not otherwise agreed upon, the licensee may use the target program in accordance with the terms of the licensing agreement whereby the Article 5(3)-techniques—which cannot be abridged pursuant to Article 9(1)—are the minimum threshold for analyzing the target program. See ULLRICH, *supra* note 76, at 75.

⁵⁸⁷ See Gesetz über Urheberrecht und verwandte Schutzrechte [Urheberrechtsgesetz] [UrhG] [Copyright Act], Sept. 9, 1965, BGBl I at 1273, available at <http://bundesrecht.juris.de>.

§ 69(g)(2) reads as follows:

“Vertragliche Bestimmungen, die in Widerspruch zu[r]...[Ausnahmeregelung für Dekompilierung in § 69(e)] stehen, sind nichtig.”

⁵⁸⁸ ULLRICH, *supra* note 76, at 76.

The Directive seems not to distinguish the status under which a party holds the program copy—either owning a copy or merely being a licensee. This view comports with the scope of the decompilation exception applying to licensees or any other persons having the right to use the copy of a program⁵⁸⁹ (such as a purchaser of a copy of a computer program). Arguably, with the bright-line rule laid down in Article 9(1) the Directive introduced predictability as regards the Article 6-right of decompilation conferred upon a licensee or an owner of a program copy since these Article 6-rights cannot be affected by contract terms—neither in shrink-wrap or click-on licenses nor contracts—that preclude copying and transforming of the original program for decompilation purposes. Nonetheless, the question remains how, if at all, this rigorous rule is to be implemented to be in accord with the freedom of contract in the EU Member States.

With the exception of the Belgian copyright statute, no other EU Member State has taken a clear position in their national copyright statute on the issue of the potential overridability of copyright exceptions by contract, beyond what is imposed by the EU Software Directives. By contrast, Article 23^{bis} of the Belgian Copyright Act⁵⁹⁰ expressly states that all the statutory exceptions granted under the Act are mandatory.⁵⁹¹ From an international perspective, the likely consequence of this statutory provision is that software providers offering their software in Belgium will try to avoid the application of Belgian law by choosing another substantive law as the governing law of the transaction, unless the application of Belgian law is made mandatory by application of protective choice of law rules (as might be the case if the user is a “consumer” under consumer protection regulations). As a result, Belgian computer users of protected software will be better protected than other users in the European Communities, which might prove problematic to the extent that it might affect the internal market of the European Communities, thus justifying the current missing harmonization of the issue at European level.

⁵⁸⁹ EU Software Directive, *supra* note 10, art. 6(1)(a), at 45.

⁵⁹⁰ Law on Copyright and Neighboring Rights of June 30, 1994.

⁵⁹¹ See Art. 23^{bis} of the Law of Aug. 31, 1998 Transposing to Belgian Legislation the European Directive No. 96/9/EC of March 11, 1996 on the Legal Protection of Databases (entered into force on Nov. 14, 1998) and amending the Belgian Copyright Act, *Moniteur belge/Belgisch Staatsblad*, 14/11/1998, p. 36.913. The document can be retrieved from the official database of the *Moniteur Belge* (Official gazette of Belgium).

1.2. Protection Under Trade Secret Law

Where the re-use of elements is not protected by copyright, nonetheless, those elements may be still subject to restrictions under other co-existing forms of protection (such as trade secret protection, protection by patent law and/or protection on competition grounds).⁵⁹² So Article 6 does, *inter alia*, not affect trade secret law.⁵⁹³ In this respect, since Article 9(1)'s first sentence leaves trade secret protection intact, it might be argued that interface information could still be retained, not on the basis of copyright, but as a trade secret.⁵⁹⁴

The second sentence of Article 9(1), which limits the effect of contractual provisions, does not apply to trade secret laws. Consequently, even if decompilation is not regarded as copyright infringement and may not be contracted out, the use of interface information might still be protected as a trade secret by way of contractual restrictions placed on the buyer or licensee of the program copy. In essence, a legitimate program user can obtain information within the limits prescribed by the Directive but he cannot not use that information. European counsels argue that such conclusion would run counter to the Directive's very purpose of guaranteeing a certain minimum of access to interface information in order to ensure interoperability.⁵⁹⁵ Therefore, Article 9(1) shall be best understood that the interface information obtained without infringement of the exclusive right may not be retained by contractual restrictions based on trade secret protection.⁵⁹⁶ This interpretation would be in accord with the U.S. legal regime, where

⁵⁹² CZARNOTA & HART, *supra* note 52, at 77, 96 (stating that cumulating both regimes of protection is possible and, therefore, if no exception is made in the Directive, the contractual clause may still be subject to challenge under a variety of national legislation or on competition grounds at a Community level).

⁵⁹³ EU Software Directive, *supra* note 10, art. 9(1), at 45, reads in full text:

“The provisions of this Directive shall be without prejudice to any other legal provisions such as those concerning patent rights, trade-marks, unfair competition, trade secrets, protection of semiconductor products or the law of contract. Any contractual provisions contrary to Article 6 or to the exceptions provided for in Article 5 (2) and (3) shall be null and void.”

⁵⁹⁴ In fact, object code is a trade secret protected by the copyright holder's exclusive rights. From the outside a user may only perceive what a computer program does but not how the program functions in the inside. In order to do so, the user must decompile the program. *See* ULLRICH, *supra* note 76, at 74.

⁵⁹⁵ Dreier, *supra* note 54, at 322, 325.

⁵⁹⁶ Spoor, *supra* note 39, at 1072-3.

decompilation is considered a fair, honest, and lawful means of uncovering a trade secret.⁵⁹⁷

U.S. software vendors on the verge of marketing their computer programs in the EU must pay attention to their software license agreements in so far as reverse engineering issues are concerned by devising them in conformance with the EU Software Directive since license agreements containing a clause that prohibits reverse engineering not only will run the risk that the single restrictive provision will be regarded to be null and void, but all the more that the entire contract will be held unenforceable.⁵⁹⁸

2. Shrink-wrap and Click-on License Agreements under U.S. Copyright Law

Opponents of anti-reverse engineering/anti-decompilation clauses rely on the pre-emption argument. State contract law that admits enforcement of contracts including anti-reverse engineering clauses is preempted by federal copyright law in either of two ways: (i) by express pre-emption, or (ii) through implied conflict pre-emption.⁵⁹⁹ Others have asserted that such clauses should be considered a misuse of copyright. Still others have suggested enforcing overbroad license terms from a freedom of contract perspective.⁶⁰⁰ Another suggestion is to enforce them unless the licensor has monopoly power.⁶⁰¹

All in all, as today, the enforceability of shrink-wrap and click-on agreements as a matter of contract law, including the questions of whether state contract law contravenes public policy and when state law can preempt federal copyright law, is unsettled.

⁵⁹⁷ See, e.g., *Kewanee Oil Co. v. Bicron Corp.*, 416 U.S. 470, 475-6 (1974) (although this case addressed the conflict between state and federal laws regarding trade secrecy, this definition is valid when applied to software reverse engineering). This notion has also been incorporated into the Uniform Trade Secret Act, the Semiconductor Chip Protection Act, and the Restatement (Third) of Unfair Competition.

⁵⁹⁸ Gushow, *supra* note 157, at 1276.

⁵⁹⁹ See III GOLDSTEIN ON COPYRIGHT, *supra* note 12, § 17.1, at 17:5.

⁶⁰⁰ For further references see Samuelson & Scotchmer, *supra* note 5, at 1628.

⁶⁰¹ Maureen A. O'Rourke, *Drawing the Boundary Between Copyright and Contract: Copyright Preemption of Software License Terms*, 45 DUKE L.J. 479, 551.

2.1. *Pre-emption by Federal Copyright Law*

2.1.1. Express Pre-emption Under Section 301(a) of the Copyright Act

Section 301(a) of the Copyright Act provides the fundamental basis for express pre-emption, saying that

all legal or equitable rights that are equivalent to any of the exclusive rights within the general scope of copyright as specified by section 106...and come within the subject matter of copyright as specified by sections 102 and 103...are governed exclusively by this title. Thereafter, no person is entitled to any such right or equivalent right in any such work under the common law or statutes of any State.⁶⁰²

Conflict pre-emption derives from the Supremacy Clause set forth in Article VI of the U.S. Constitution.⁶⁰³ Conflict pre-emption occurs when either the federal and state laws directly conflict, so that it is physically impossible for a party to comply with both, or a state law creates an obstacle to the accomplishment and execution of objectives that Congress envisioned upon creation of the federal law.⁶⁰⁴ Applied to the decompilation context, conflict pre-emption may become an issue if enforcing a contractual restriction on decompilation has the effect of restricting the reach of the fair use defense or section 1201(f) of the DMCA.⁶⁰⁵ To the extent that click-wrap or click-on agreements are determined to be binding under state law, state law effectuates to regulate the same subject matter as does federal copyright law. However, judiciary has inferred an exception to the strict pre-emption concept of the Copyright Act, the so-called “additional elements” test.

In *Bowers v. Baystate Technologies, Inc.*, the Federal Circuit upheld a contractual provision in a computer software license that waived reverse engineering of the

⁶⁰² 17 U.S.C. § 301(a) (2000).

⁶⁰³ The second clause in Article VI of the U.S. Constitution declares that the Constitution, all laws made in furtherance of the Constitution, and all treaties made under the authority of the United States are the “supreme law of the land” and enjoy legal superiority over any conflicting provision of a state constitution or law.

U.S. CONST. art. VI, cl. 2, reads in full text:

“This Constitution, and the Laws of the United States which shall be made in Pursuance thereof; and all Treaties made, or which shall be made, under the Authority of the United States, shall be the supreme Law of the Land; and the Judges in every State shall be bound thereby, any Thing in the Constitution or Laws of any State to the Contrary notwithstanding.”

⁶⁰⁴ See, e.g., *Davidson & Associates v. Jung*, 422 F.3d at 638 (8th Cir. 2005).

⁶⁰⁵ See Merritt A. Gardiner, *Bowers v. Baystate Technologies: Using the Shrinkwrap License to Circumvent the Copyright Act and Escape Federal Preemption*, 11 U. MIAMI BUS. L. REV. 105, 110-1 (2003).

purchased software, finding the reverse engineer liable for copyright infringement and breach of contract under Massachusetts contract law.⁶⁰⁶ Harold L. Bowers (“Bowers”) developed and patented software to improve the usability of already existing computer-aided design (CAD) software, “Designer’s Toolkit”. Bowers distributed his software with a shrink-wrap license, *inter alia*, prohibiting any reverse engineering.⁶⁰⁷ Subsequently, Baystate Technologies, Inc. (“Baystate”), a competitor to Bowers, developed and marketed “Draft-Pak”, a similar software incorporating many of the features contained in Bowers’ Designer’s Toolkit.⁶⁰⁸ Increased price competition between the Designer’s Toolkit and Draft-Pak followed Baystate’s introduction of Draft-Pak. Baystate eventually sued Bowers for declaratory judgment, thereupon Bowers counterclaimed for copyright infringement, patent infringement and breach of contract.⁶⁰⁹ In relying on the frequently cited case *ProCD, Inc. v. Zeidenberg*,⁶¹⁰ the Federal Circuit supported its reasoning that the mutual assent and consideration required by a contract claim constitute “additional elements” that render the state law claim qualitatively different from copyright infringement.⁶¹¹ Furthermore, the court reiterated Judge Easterbrook’s opinion in *ProCD*⁶¹² that contract rights are not equivalent to the rights conferred upon parties by the Copyright Act because copyrights create exclusive rights “against the world”, while shrink-wrap licenses are simple two-party transactions that only affect their parties and cannot bind a non-party.⁶¹³ Consequently, the court emphatically rejected Baystate’s express preemption claim.

⁶⁰⁶ *Bowers v. Baystate Technologies, Inc.*, 320 F.3d 1317, 1326 (Fed. Cir. 2003) (Dyk, J., concurring in part, dissenting in part), *cert. denied*, 539 U.S. 928 (2003) [hereinafter Bowers].

⁶⁰⁷ 320 F.3d at 1322.

⁶⁰⁸ *Id.*

⁶⁰⁹ *Id.*

⁶¹⁰ *ProCD, Inc. v. Zeidenberg*, 86 F.3d 1447 (7th Cir. 1996) (holding shrink-wrap license governing the use of computer software not to be preempted by the Copyright Act in a Wisconsin contract law analysis) [hereinafter *ProCD*].

Additionally, the court referred to the law of other circuits to demonstrate that the Copyright Act does not preempt contractual constraints on copyrighted articles, namely *Wrench LLC v. Taco Bell Corp.*, 256 F.3d 446 (6th Cir. 2001); *Nat’l Car Rental Sys., Inc. v. Computer Assocs. Int’l, Inc.*, 991 F.2d 426 (8th Cir. 1993); *Taquino v. Teledyne Monarch Rubber*, 893 F.2d 1488 (5th Cir. 1990); *Acorn Structures v. Swantz*, 846 F.2d 923 (4th Cir. 1988).

⁶¹¹ *Bowers*, 320 F.3d at 1324-5.

⁶¹² *ProCD*, 86 F.3d at 1454.

In addition, the First Circuit cited *Data General Corp. v. Grumman System Support Corp.*, 36 F.3d 1147, 1168 (1st Cir. 1994) (holding that the Copyright Act did not preempt the state law trade secret claim if additional elements of proof such as a trade secret and breach of a duty of confidentiality were qualitatively different from a copyright claim).

⁶¹³ *Bowers*, 320 F.3d at 1325.

The rule deducible from these precedents goes that the plaintiff's attempt to invalidate a shrink-wrap license provision containing a no-reverse engineering clause on express pre-emption grounds will go awry because the suit will always contain the additional elements pertaining to a valid contract.⁶¹⁴ This rule provides freedom of contract with much respect.⁶¹⁵

However, proponents of the express pre-emption argument under section 301(a) object the rule on three counts. First, the legislative history of the Copyright Act buttresses the equivalence of contractual provisions prohibiting reverse engineering to those protected by the Copyright Act. The legislative records state that Congress' intent was that "[a]s long as a work fits within one of the general subject matter categories of sections 102 and 103, the bill prevents the States from protecting it even if it fails to achieve Federal statutory copyright [protection]."⁶¹⁶ This leads inevitably to the second flaw inherent in the rule. In the past, software decompilation was held fair use.⁶¹⁷ Even though acts of decompilation are not protected by the Copyright Act, Congress would authorize federal pre-emption of licensing provisions that prohibit it. In *Bowers*, the Federal Circuit's majority opinion failed to recognize the doctrine of fair use,⁶¹⁸ and thereby undermined the federal Copyright Act. Neither did the court consider the copyright misuse doctrine. Third, the reasoning in *Bowers* is incomplete as the majority opinion missed out to apply the test for implied conflict pre-emption under the Supremacy Clause of the U.S. Constitution.⁶¹⁹

2.1.2. Conflict Pre-emption Under the Supremacy Clause

In *Vault*, the Fifth Circuit, in applying Louisiana law, held a state law that is prohibiting all copying of a computer program, including intermediate copying as a

⁶¹⁴ Pruitt, *supra* note 374, at 66; *but see* 1 NIMMER ON COPYRIGHT, *supra* note 15, § 1.01[B][1][a][iii], at 1-21-2 (2006) (criticizing that the Seventh Circuit in *ProCD* reached a overly broad conclusion given the fact-specific holdings of the preceding cases on that issue).

⁶¹⁵ *See also* Seungwoo Son, *Can Black Dot (Shrinkwrap) Licenses Override Federal Reverse Engineering Rights?: The Relationship Between Copyright, Contract, and Antitrust Laws*, 6 TUL. J. TECH. & INTELL. PROP. 63, 126-7 (2004).

⁶¹⁶ H.R. REP. NO. 94-1476, at 129-31(1976), *reprinted in* 1976 U.S.C.A.A.N. 5747.

⁶¹⁷ *See cases cited supra* Part IV.

⁶¹⁸ 4 NIMMER ON COPYRIGHT, *supra* note 15, § 13.09[A][3][b], at 13-303 (2006) (stating that the decision did not address the merits of any claim for copyright infringement).

⁶¹⁹ *See* Gardiner, *supra* note 605, 108-11.

result of the decompilation process, is preempted by the federal Copyright Act.⁶²⁰ As dealt with in greater detail elsewhere in this paper,⁶²¹ the pertaining facts here are outlined as follows. Vault, a seller of computer software, brought an action seeking to enjoin Quaid, a software manufacturer, from selling a program that unlocked Vault's computer software program and thus permitted the making of copies. Quaid had allegedly purchased Vault's software and decompiled it to develop, subsequently, its own unlocking program. Vault, which sold its software with a shrink-wrap agreement, claimed that Quaid's actions violated the Louisiana Software License Enforcement Act and the contract restrictions that were consistent with that Act. The court determined that the Louisiana Software License Enforcement Act was preempted by federal law because it conflicted with a portion of the Copyright Act⁶²² and thus "'touched upon the area' of federal copyright law."⁶²³ Thus the restrictions in the shrink-wrap agreement that relied upon the provisions of Louisiana state law were unenforceable.

More recently, however, in *Davidson & Associates v. Jung*,⁶²⁴ the Eight Circuit dismissed the defendants' reliance on *Vault* and concluded that Congress did not intend to strip private parties of their right to freely negotiate contractual terms prohibiting reverse engineering.⁶²⁵ The plaintiff company ("Blizzard") was a software provider that developed and sold computer games for use on personal computers. To further the sales of its computer game software, Blizzard developed an online network called "Battle.net" that exclusively allowed purchasers of its

⁶²⁰ *Vault*, 847 F.2d 255, 270 (5th Cir. 1988).

⁶²¹ See discussion *supra* Part IV.1.2.

⁶²² The court noted numerous conflicts between the Louisiana Software License Enforcement Act and the federal Copyright Act, including: (1) while the Louisiana License Act authorizes a total prohibition on copying, the Copyright Act allows archival copies and copies made as an essential step in the utilization of a computer program, 17 U.S.C. § 117; (2) while the Louisiana License Act authorizes a perpetual bar against copying, the Copyright Act grants protection against unauthorized copying only for the life of the author plus fifty years, 17 U.S.C. § 302(a); and (3) while the Louisiana License Act places no restrictions on programs which may be protected, under the Copyright Act, only "original works of authorship" can be protected, 17 U.S.C. § 102. See *Vault*, 847 F.2d at 269.

⁶²³ 847 F.2d at 269.

⁶²⁴ *Davidson Associates, Inc. v. Internet Gateway*, 334 F.Supp.2d 1164 (E.D. Mo. 2004), *aff'd sub nom. Davidson & Associates v. Jung*, 422 F.3d 630 (8th Cir. 2005) [hereinafter *Jung*].

⁶²⁵ See 422 F.3d at 639 (citing *Bowers*, 320 F.3d at 1325-6 (Dyk, J., dissenting) (opining that although parties may properly negotiate away their statutory rights, the shrink-wrap license enveloping the Bowers' CAD software was a contract of adhesion so that agreeing to its terms did not involve true negotiation)).

software to play against other purchasers through the network.⁶²⁶ In order for users to log on to Battle.net, Battle.net used a “secret handshake” to allow only authenticated, i.e. not pirated, versions of Blizzard’s gaming software to be played on its multi-game platform.⁶²⁷ Simultaneously when installing a Blizzard gaming software, a user had to agree to a click-wrap license agreement that prohibited reverse engineering of Blizzard’s software for any reason.⁶²⁸ In the wake of experiencing problems with Battle.net, the defendant (“Jung”) and other Blizzard game subscribers engaged in acts of decompilation to develop an open-source website, “bnetd.org”, which emulated the experience of playing Blizzard games on Battle.net.⁶²⁹ Blizzard sued, alleging, *inter alia*, copyright infringement, a violation of the DMCA’s anti-circumvention and anti-trafficking provisions and breach of the click-wrap license agreement. The defendant appealed, disputing a DMCA violation and contending that conflict pre-emption precluded state enforcement of the anti-reverse engineering licensing provisions.⁶³⁰ The Eight Circuit held that the rights of private citizens to contract neither conflicts with the reverse engineering exception under section 1201(f) of the DMCA⁶³¹ nor restricts rights under federal law.⁶³² With favorable reference to *Bowers*, the court held that private parties may contract around default copyright limitations,⁶³³ which was exactly what the users in the case at bar did by “signing”⁶³⁴ the click-on license by which they expressly

⁶²⁶ *Id.* at 633.

⁶²⁷ *Id.*

⁶²⁸ *Cf. id.* at 634-5 (citing the exact wording of the End User License Agreement and the Terms of Use).

⁶²⁹ *Id.* at 635.

⁶³⁰ *Id.* at 637.

⁶³¹ The court found defendants’ circumvention constituted infringement. The purpose of the bnetd.org platform was to provide an emulated online environment for users of Blizzard games who wanted to play in a multi-player environment without using a valid CD key to enter Battle.net. The bnetd.org emulator did not determine whether the CD key was valid or currently in use by another player. As a result, unauthorized copies of the Blizzard games were freely played on bnetd.org servers. Given this finding, the defendant failed to establish a genuine issue of material fact as to the applicability of the interoperability exception of section 1201(f). *See id.* at 639.

⁶³² *Id.*

⁶³³ *Id.*

⁶³⁴ The manifestation of assent by clicking on “I agree” has been acknowledged by courts in a variety of contractual settings. *See, e.g., Hughes v. McMenamon*, 204 F.Supp.2d 178 (D. Mass. 2002) (internet service agreement); *Moore v. Microsoft Corp.*, 741 N.Y.S.2d 91 (N.Y. App. Div. 2002) (purchase of computer software); *Forrest v. Verizon Communications, Inc.*, 805 A.2d 1007, 1011 (D.C. Ct. App. 2002) (stating that “[a] contract is no less a contract simply because it is entered into via a computer”).

relinquished their rights to decompilation.⁶³⁵ In effect, the court held that fair use privileges can be waived and that contractual claims are not governed by copyright law. The purpose and appropriate function of contracting in any property-rights arena is to tailor the parties' relationship in a manner different from that set out in property law. So contract and intellectual property law were to involve entirely different concepts. The federal Copyright Act contains certain background provisions, such as fair use, which yet are not clearly identified as default or mandatory rules.⁶³⁶

The crux for courts lies with finding the right balance between the rights of individuals to contract and the public policy promoted through copyright. Two tools can be employed by the courts to limit contractual provisions in software license agreements that prohibit reverse engineering: (i) the fair use defense, and (ii) the copyright misuse defense. So far courts have inclined not only to defy reverse engineering as fair use but also have negated to hold prohibitions on reverse engineering to be a misuse of copyright.

2.2. *Restrictive Licensing Practices as Copyright Misuse*

Another avenue by which overreaching restrictive license terms in shrink-wrap and click-on license agreements might be voided is by claims of antitrust violation (under tying or monopolization theories) or by asserting the equitable defense of copyright misuse.⁶³⁷ Courts have traditionally found misuse in situations where the copyright owner either has entered into anti-competitive licensing agreements or extended its monopoly beyond the scope of the copyright or violated the public policies underlying copyright laws. Recently courts have begun to embrace a broader misuse doctrine by adopting the latter approach.⁶³⁸ Certainly, the prohibition on reverse engineering keeps the licensee from accessing the functional

⁶³⁵ *Jung*, 422 F.3d at 639 (citing *Bowers*).

⁶³⁶ O'Rourke, *supra* note 601, at 480 (1995).

⁶³⁷ See generally *Lasercomb America, Inc. v. Reynolds*, 911 F.2d 970 (4th Cir. 1990); for a discussion of the U.S. doctrine of copyright misuse see also *infra* Part VII.1.2.

⁶³⁸ See *Video Pipeline, Inc. v. Buena Vista Home Entertainment, Inc.*, 342 F.3d 191, 206 (3d Cir. 2003) (opining that copyright misuse might operate beyond its traditional anti-competition context in a meaning to abstract the misuse of copyright from the traditional anti-competition context developed in patent misuse cases).

elements of the computer program code, thus arguably extending the scope of the limited monopoly over the copyrighted expression beyond the scope of copyright law. Therefore, such a contract clause might undermine the public policy underlying copyright law and so constitutes copyright misuse.⁶³⁹

The nature of decompilation makes courts walk the thin line between copyright policy considerations and economic consequences for the involved, including the general public. Given that challenge for copyright law regimes all over the world the misuse doctrine in balancing copyright policy and net-effects on competition⁶⁴⁰ is capable of contributing an alternative tool in the software decompilation battle. Nonetheless, the courts are still reluctant to take the copyright misuse and its valuable contribution to solving the decompilation dilemma for granted.⁶⁴¹ How had the case of *Bowers* ended up had Baystate invoked the doctrine of copyright misuse?

With the balancing effect of the misuse doctrine, one scarcely may understand the courts' suspicions of the doctrine's merits. The Federal Circuit in *Bowers* stated that its holding on pre-emption did not affect its position that reverse engineering is a fair use exception to copyright infringement.⁶⁴² Even if the contractual prohibition on reverse engineering had not been enforceable, it is likely that Baystate's attempt at invoking the fair use defense would have failed. Primarily, the first factor, the purpose and character of the use, would prevail the fair use analysis and ultimately be weighed in favor of Bowers. Baystate's use of the reverse-engineered product to create a commercial product, Draft-Pak, that would closely match the functionality of Bower's software, Designer's Toolkit, would have difficulties to satisfy the transformation threshold under the *Sony* rule. In addition, the fourth factor, the effect of Baystate's use of Bowers' software

⁶³⁹ See Jeffrey A. Andrews, *Reversing Copyright Misuse: Enforcing Contractual Prohibitions On Software Reverse Engineering*, 41 HOUS. L. REV. 975, 1000-1 (2004).

⁶⁴⁰ In the past, disparate case law has shown the difficulties for courts to delineate precisely between antitrust violation and misuse. See, e.g., *In re Napster, Inc. Copyright Litigation*, 191 F.Supp. 2d 1087, 1105 (N.D. Cal. 2002) (discussing cases and coming up with a modified public policy approach which considered copyright policy as well as the net-effect on competition).

⁶⁴¹ The Supreme Court has never affirmed explicitly the applicability of a copyright misuse defense. The Supreme Court, however, implicitly has recognized the existence of the copyright misuse doctrine. See, e.g., *United States v. Paramount Pictures, Inc.*, 334 U.S. 131 (1948); *United States v. Loew's, Inc.*, 371 U.S. 38 (1962); *Broadcast Music, Inc. v. Columbia Broadcasting System*, 441 U.S. 1 (1979); *Lasercomb*, 911 F.2d at 976.

⁶⁴² *Bowers*, 320 F.3d at 1325.

functionality upon the potential market for or value of Bowers' software, would weigh heavily in Bowers' favor. Evidence indicated that the newly released Draft-Pak, which included much of the Designer's Toolkit underlying functions, was having a measurable effect on the market for Designer's Toolkit. In conclusion, Baystate would have had no success in the litigation by relying on fair use. However, rejecting the fair use defense to infringement would not be the end of the day for Baystate as Baystate, alternatively, could have invoked the copyright misuse doctrine.⁶⁴³ Generally, the facts in *Bowers* could give rise to a copyright misuse analysis. The shrink-wrap license agreement excluded Designer's Toolkit's users from any engagement in reverse engineering.⁶⁴⁴ Because reverse engineering is the only process by which non-copyrightable functions can be accessed from object code, the licensing agreement in fact prohibited access to non-protectible elements underlying Bowers' copyrighted computer program and thus extended the scope of Bowers' copyright beyond its borders granted by copyright. Inexplicably, the copyright misuse defense was not even raised in the Federal Circuit.

In the recent case *Altera Corp. v. Clear Logic, Inc.*,⁶⁴⁵ addressing semiconductor chip infringement, the Ninth Circuit held that neither pre-emption⁶⁴⁶ nor misuse law precludes the enforcement of a valid contract that prohibits the use of a chip bitstream to reverse engineer design.⁶⁴⁷ Altera customers were licensed to use Altera's copyrighted software only to program Altera's chip.⁶⁴⁸ The defendant ("Clear Logic") argued that Altera's license agreements should not be enforced because they operate as illegal tying arrangements. In other words, Altera was alleged to (mis)use its copyright in its programming software to control competition in an area outside its copyright granted (i.e. Altera's chip hardware).⁶⁴⁹

⁶⁴³ Andrews, *supra* note 639, at 1003.

⁶⁴⁴ *See id.* (speculating that Baystate might have taken into account that the defense could not prevail before the Federal Circuit because unlike the Courts of Appeal for the Fourth, Fifth, and Ninth Circuits, the Federal Circuit has clearly articulated its position on the misuse defense in the patent context and accordingly, copyright misuse should be considered in light of an antitrust approach, directed by a rule of reason analysis (citing *Mallinckrodt, Inc. v. Mediapart, Inc.*, 976 F.2d 700, 706 (Fed. Cir. 1992)).

⁶⁴⁵ *Altera Corp. v. Clear Logic, Inc.*, 424 F.3d 1079 (9th Cir. 2005) [hereinafter *Altera*].

⁶⁴⁶ The Ninth Circuit found the "sole use" provision in the computer software license is an "additional element" which saves the state law claim from pre-emption. *See Altera*, 424 F.3d at 1089-90.

⁶⁴⁷ 424 F.3d at 1082.

⁶⁴⁸ *Id.*

⁶⁴⁹ *Id.* at 1090.

Nonetheless, the court rejected Clear Logic’s copyright misuse argument by holding that there has been no allegation of copyright infringement and therefore invoking the defense to copyright misuse was mistaken. The court stated that “[c]opyright misuse is not a defense to the state law claims asserted by Altera.”⁶⁵⁰ The courts apparently are to continue to disregard copyright misuse as an effective tool in the hands of reverse engineers against software licensors misusing their copyright. It is a victory of the private ordering system over the public ordering system (federal statutes) at the expense of not only the reverse engineers but the copyright policy as well.

2.3. *The Doctrine of Unconscionability*

Courts preferring freedom of contract to pre-emption and consequently permitting software providers to contract away a fair use defense regularly omitted to determine in a second step whether the contract was freely and fairly negotiated and thus did not meet the threshold of unconscionability. Providers of mass-market software do not distinguish between purchasers who will decompile for a legitimate reason and those who will do so for developing competing products—simply because individually negotiated transactions are impractical in the mass market.⁶⁵¹ Provisions against decompilation in software license agreements seem overbroad because there often exist legitimate, non-infringing reasons why one might seek access to the underlying source code. For instance, one might think of research or educational purposes or error correction. This aspect leads inevitably to the notion of unconscionability.

The unconscionability doctrine seeks mainly to protect against contracts that are grossly one-sided due to superior bargaining power, or present oppression and unfair surprises to consumers. The basic test is whether a term or contract involved is so one-sided as to be unconscionable under the circumstances existing at the time of the making of the contract.⁶⁵² Unconscionability may occur in a procedural and

⁶⁵⁰ *Id.*

⁶⁵¹ O’Rourke, *supra* note 601, at 494-6 (1995) (demonstrating in general the contracting process between software providers and software purchasers).

⁶⁵² U.C.C. § 2-302, Official Comment Para. 1 (West 2006).

substantive form.⁶⁵³ In order to invalidate a contract, both forms must be shown on a sliding scale; i.e., the more procedurally unconscionable a contract, the less substantively unconscionable it needs to be and vice versa.⁶⁵⁴ As shrink-wrap and click-on licenses are often presented on a take-it-or-leave-it negotiating basis, software providers usually are willing to license their software only under their terms and conditions. The element of adhesion in addition to generally unsophisticated subscribers to a shrink-wrap or click-on license agreement will prompt courts taking an unsympathetic view towards the software provider.⁶⁵⁵ But even where procedural unconscionability exists, the subscriber still must show that the anti-reverse engineering term is substantively unconscionable. For example, in *Davidson & Associates, Inc. v. Internet Gateway*, the court held an anti-reverse engineering clause not a harsh or oppressive term.⁶⁵⁶ Furthermore, even under state trade secret law, it is not a misappropriation to discover or appropriate a trade secret by reverse engineering.⁶⁵⁷ Nevertheless, licensees may sway the court's view by arguing that prohibiting reverse engineering in the license agreement effectively prevents the user from detecting and fixing bugs that could be corrected if the user had access to the program's source code.⁶⁵⁸ Software vendors may avert such substantive unconscionability concerns in advance by restricting their license terms to decompilation in lieu of the broad concept of reverse engineering in general. While prohibiting reverse engineering in general deprives the licensee of access to the software's functionality altogether, by prohibiting solely decompilation the licensee still may engage in other forms of reverse engineering, such as black-box

⁶⁵³ See, e.g., Son, *supra* note 615, at 121-2.

⁶⁵⁴ See *Comb v. Paypal, Inc.*, 218 F.Supp.2d 1165, 1172 (N.D. Cal. 2002).

⁶⁵⁵ Cf. *Davidson & Associates, Inc. v. Internet Gateway*, 344 F.Supp.2d 1164, 1179 (E.D. Mo. 2004) (rejecting an unconscionability argument on grounds that the subscribers to the licensing agreement were not "unwitting members of the general public, but were computer programmers familiar with the technical language used in the licensing agreement, and had the expertise to reverse engineer and understand source code").

⁶⁵⁶ See 344 F.Supp.2d at 1180.

⁶⁵⁷ See RESTATEMENT (THIRD) OF UNFAIR COMPETITION § 43 (1995): "... Independent discovery and analysis of publicly available products or information are not improper means of acquisition."

See also Anthony J. Mahajan, *Intellectual Property, Contracts, and Reverse Engineering After ProCD: A Proposed Compromise for Computer Software*, 67 Fordham L. Rev. 3297, 3318 (1999) (trade secret protection strikes the balance between a stronger protection of source code by security measures, such as password access and encryption, while still granting the general public access to the functional elements underlying a computer program; if shrink-wrap licenses are enforceable, the delicate balance once tipped will be shattered).

⁶⁵⁸ Pruitt, *supra* note 533, at 66.

analysis and thereby gaining access to the unprotectible aspects of the target program. Such a narrow drawn clause in a shrink-wrap or click-on license may read in part to prohibit “decompilation to reverse engineer infringing products.” After all, software licensees will face difficulties to satisfy the criteria of unconscionability of anti-reverse engineering provisions contained in shrink-wrap or click-on license agreements given the seemingly present attitude pursued in U.S. courts to limit the decompilation rights to a minimum.⁶⁵⁹

VII. A Transatlantic Comparison In Re Software Decompilation

1. *What Happened So Far: Fair Use, Copyright Misuse and Article 6 of the EU Software Directive*

1.1. Is the Fair Use Concept Really Fair?

The legal fundament of most EU Member States rests on civil law traditions. Common law devices, such as the fair use or the copyright misuse defense, are not known in most of the EU Member States. Also the EU Software Directive refrained from legal treatment of decompiling computer programs under fair use or copyright misuse analyzes. Article 6 of the EU Software Directive mandates permitting software decompilation under narrow statutory circumstances. At the early stage of the law evolving on the decompilation issue, Article 6 appeared to introduce a stricter approach than the Anglo-American fair use/fair dealing concept.⁶⁶⁰ Under the Directive, the legal basis for a right to access of information embedded in computer software is the idea/expression dichotomy (i.e., ideas, principles and methods of operation are not copyrightable) and the right of a legitimate owner of a program copy to use the program that includes lawful analyzing of the original program through certain acts enumerated in the Directive.

Across the Atlantic, federal jurisdiction tried to settle the decompilation battle via fair use tactic. Software decompilation, usually followed by reverse engineering the computer program previously decompiled, frequently has occurred as fair use

⁶⁵⁹ *See id.*

⁶⁶⁰ *Cf. CZARNOTA/HART, supra note 52, at 126.*

defense to copyright infringement claimed by the lawful program owner.⁶⁶¹ First, the decompiled target program is subject to the abstraction-test in order to filter out the copyrighted core of the program. Then, there must have been intermediate copying for analyzing purposes. Lastly, the copying will be subject to weighing of the factors of 17 U.S.C. § 107—regardless of whether the reverse engineered program shows substantial similarities or not.

Compared with the purpose of Article 6 of the EU Software Directive, the fair use analysis appears not to be limited to, but goes beyond, the disclosure of interface information for interoperability purposes.⁶⁶² U.S. courts only allowed reverse engineering, including decompiling, computer code provided that two requirements are met. First, decompilation must be undertaken for a “legitimate reason” which may theoretically mean more than interoperability. Second, decompilation must provide the only means of access to those elements of the code that are not protected by copyright.⁶⁶³ In spite of the clearly worded rule the fair use defense construed as equitable rule of reason,⁶⁶⁴ especially in the reverse engineering context, faces the lack of bright-line rules and unpredictable, non-traditional fair use analyses by courts intermingling their sense of fairness with copyright policy into analyses of intermediate copyists’ infringement.⁶⁶⁵ As consequence of that flexibility, early case law on its face showed U.S. courts to side with reverse engineers⁶⁶⁶ in furtherance of the public dissemination of new software which, in turn, will promote the constitutional goal of advancing technology and intellectual growth.⁶⁶⁷

⁶⁶¹ In *Sega* and *Sony*, the Ninth Circuit stripped of the copyright incentives and rewards that first coaxed software developers into producing and disseminating their works. *See* case discussions *supra* Parts IV.2.3. et seq.

⁶⁶² ULLRICH, *supra* note 76, at 82-3.

⁶⁶³ *Sega*, 977 F.2d at 1518.

⁶⁶⁴ Under an “equitable rule of reason” principle, a court decides a case according to its perception of what a fair outcome would be based on the case’s specific facts. *See* H.R. REP. NO. 94-1476, 94th Cong., 2d Sess. 65 (1976).

⁶⁶⁵ Lewis, *supra* note 91, at 574-5, 579 (applying the fair use doctrine by U.S. courts in a too arbitrary and thus unpredictable extent may discourage software developers and thereby resulting in counter-effective outcomes).

⁶⁶⁶ Williams, *supra* note 445, at 272 (concluding that the *Sega* court seemingly molded its fair use analysis to support a predetermined outcome, and thus, did not give serious regard to copyright law’s objective).

⁶⁶⁷ U.S. CONST. art. 1, § 8, cl. 8.

Furthermore, unlike the provisions of Article 6(2) of the EU Software Directive, U.S. case law as for the time being has not ruled on what may be done with the results of the decompilation and reverse engineering. In applying the fair use doctrine, U.S. courts negligently discarded the intent obviously pursued by reverse engineering defendants by characterizing their purpose as legitimate and essentially non-exploitative.⁶⁶⁸ Courts simply accepted the defendants' assertion that the study of functional requirements is solely based on the incentive to achieve interoperability. Here the inquiry ended. The judges closed their eyes to the ultimate aim, which is to develop programs based on the decompiled result that would directly compete with the plaintiffs' existing software, and therewith an aim that undoubtedly was directed to commercial exploitation.⁶⁶⁹ Following the lead in *Sony*, litigants will likely continue to attribute significant weight to the non-infringing, thus transformative, nature of the product, particularly given the fact that copyright infringement cases are often settled at the preliminary injunction stage—a stage at trial, when litigants are still unable to furnish the court with sufficient evidence to allow for a meaningful analysis of the economic loss factor.⁶⁷⁰ Software innovators who have invested significant resources in the developing and creation of their computer programs will end up in frustration and that, eventually, will run counter to the copyright policy of encouraging the “artistic creativity for the general public good.”⁶⁷¹ As today, internationally competing software developers are free to invest in expensive software development and subsequently to enter emerging markets. In so doing, they will capitalize on more bright-line laws on decompilation, thereby gaining a competitive advantage in the software market vis-à-vis U.S. software developers.⁶⁷²

Evidently, in many respects the U.S. approach is at odds with the Continental approach. In an attempt to align the adaptable Anglo-American notion of fair use/fair dealing with the Continental civil law tradition, which gives legal certainty

⁶⁶⁸ Research, particularly in the form of decompilation, is rarely, if ever, performed for research's sake unless there is some prospect of financial gain and commercial success to be expected. *See Campbell*, 510 U.S. at 584.

⁶⁶⁹ *See, e.g., Sega*, 977 F.2d at 1522, 1526 (by gaining interoperability, Accolade wanted to directly compete against Sega and prosper in the Genesis videogame market, which it succeeded in doing).

⁶⁷⁰ Rothman, *supra* note 97, at 7.

⁶⁷¹ *See Sega*, 977 F.2d at 1527; *see also Williams*, *supra* note 445, at 267-8.

⁶⁷² *Id.*, at 283.

ab initio as to what is permitted and what is excluded, the general clause of Article 6(3) thus was added to the EU Software Directive.⁶⁷³ By relying on Article 6(3), as sort of a hatch,⁶⁷⁴ European Communities' judges are given a statutory resort to prohibit those acts of decompilation which would be proscribed under the doctrine of fair use even though principally in line with the wording of Article 6(1) and (2).⁶⁷⁵ However, there is still the ongoing copyright policy clash which makes it all the more difficult to align the copyright law regimes of both jurisdictions.

1.2. Copyright Misuse As A Fairer Legal Basis For Software Decompilation Under U.S. Law?

Interface specifications fall, at least partly, in the category of ideas and functional concepts and therefore are unprotected by copyright.⁶⁷⁶ Thus, in addition to the fair use doctrine, the copyright misuse doctrine is a further safety valve available under U.S. copyright law in cases involving decompilation.⁶⁷⁷

1.2.1. The Doctrine of Intellectual Property Misuse

Intellectual property misuse, a judicial doctrine, arises when a holder of an intellectual property right uses that right to obtain or coerce an unfair commercial advantage beyond the scope of the right.⁶⁷⁸ Under the traditional formulation, misuse renders an intellectual property unenforceable when the owner attempts to

⁶⁷³ For a discussion *see supra* Part III.3.3.3.

⁶⁷⁴ See Haaf, *supra* note 23, at 424 (identifying Article 6(3) as a “counterweight to the Article 6’s potentially overly liberal grant of permission to decompile programs for purposes of interoperability”).

⁶⁷⁵ See LEHMANN, *supra* note 2, at 24.

⁶⁷⁶ See Rothman, *supra* note 97, at 8 (stating that the copyright misuse doctrine in connection with interface specifications appears to be a more effective means in furtherance of unrestricted access to and use of unprotected ideas and functional concepts than the fair use doctrine); *see also* Lewis, *supra* note 91, at 581-6.

⁶⁷⁷ Arguably, Sony was attempting to use the copyright in its BIOS to obtain a monopoly over unpatented and uncopyrightable interface specifications in order to prevent the development of a software product that would emulate and thus compete with the unpatented and uncopyrighted hardware components of its PlayStation. The Ninth Circuit, however, did not take the opportunity to clarify the application of this doctrine to software reverse engineering. *See id.*

⁶⁷⁸ To give the policies underlying intellectual property law teeth, courts have refused to enforce otherwise valid patents and copyrights when an infringement defendant establishes that the plaintiff has “misused” its legal monopoly. *See, e.g., Alcatel U.S.A., Inc. v. DGI Tech., Inc.*, 166 F.3d 772, 792 (5th Cir. 1999).

extend the statutory monopoly beyond its lawful scope or otherwise engages in conduct contrary to the public policy inherent in the intellectual property statute.⁶⁷⁹

If the alleged infringer is able to show misuse, the right is held unenforceable until the anticompetitive effect of the misuse has been purged.⁶⁸⁰ Courts most frequently have found misuse in cases of tying arrangements⁶⁸¹ and restrictions on the production or sale of competing goods.⁶⁸²

Intellectual property misuse doctrines grew from the equitable notion that a plaintiff with “unclean hands”⁶⁸³ could not use the courts to obtain redress. Formulating and applying misuse principles allows the courts to balance the equities between the parties to a case while safeguarding the public policy underlying the governing statutory scheme.

Since the copyright misuse defense is, hitherto, an unwritten law, courts are provided with discretion in a broad province. It is just the federal common-law-making power which enables courts to deal with the interdependence between competition, copyright, and patent law inherent in the balancing effect of a copyright misuse analysis.⁶⁸⁴ In a “balance of equities”⁶⁸⁵ the copyright misuse defense provides an effective tool for gap-filling and coordinating related and interdependent bodies of law, i.e., whenever those other remedies found in

⁶⁷⁹ See, e.g., *Morton Salt Co. v. G.S. Suppiger Co.*, 314 U.S. 488, 492-4 (1942) [hereinafter *Morton Salt*].

⁶⁸⁰ See *Morton Salt*, 314 U.S. at 493 (noting a misuser cannot sue for infringement of its patent “until it is made to appear that the improper practice has been abandoned and that the consequences of the misuse of the patent have been dissipated”); *Practice Mgmt. Info. Corp. v. American Med. Ass’n.*, 121 F.3d 516, 520 n.9 (9th Cir. 1997) (reiterating that “[c]opyright misuse does not invalidate a copyright, but precludes its enforcement during the period of misuse”).

⁶⁸¹ See *id.*, at 189-94.

⁶⁸² See, e.g., *Lasercomb Am., Inc. v. Reynolds*, 911 F.2d 970, 972-3 (4th Cir. 1990).

⁶⁸³ The unclean hands doctrine, as an equitable doctrine with historical roots in the medieval times, evolved from the discretionary nature of equitable relief in English courts of equity, such as Chancery. See, e.g., *State ex rel. Summa v. Starke Circuit Court*, 238 Ind. 204, 209 n.1 (149 N.E.2d 541) (Ind. 1958) (stating that the maxim that ‘he who comes into equity must come with clean hands’ is an ancient and favorite precept of the chancery court).

⁶⁸⁴ But see Brett Frischmann & Dan Moylan, *The Evolving Common Law Doctrine of Copyright Misuse: A Unified Theory and Its Application to Software*, 15 BERKELEY TECH. L.J. 865, 876 (2000) (arguing that this balance should be left exclusively to Congress because of its constitutional authority under the Intellectual Property Clause and its presumptive institutional competence in developing policy, while the role of judges should be limited to coordinate, or at least to signal to Congress that coordination is necessary, where friction between the two intellectual property regimes arises).

⁶⁸⁵ *Id.*, at 877.

competition law, the fair use doctrine, or statutory creation are legally inapplicable or practically undesirable.⁶⁸⁶

The origin of the copyright misuse defense is said to lie in the specialized doctrine of patent misuse⁶⁸⁷ created by the courts to restrain anticompetitive abuses of the patent monopoly.

1.2.2. Pre-Lasercomb: The Patent Misuse Doctrine

1.2.2.1. Morton Salt Co. v. G.S. Suppiger Co.

The judicial creation of intellectual property misuse doctrines began with patent misuse as early as 1917.⁶⁸⁸ But it is the Supreme Court's decision in *Morton Salt Co. v. G.S. Suppiger* in 1942 that is considered the leading patent misuse case.⁶⁸⁹

Morton Salt involved a tying arrangement requiring purchase of an unpatented item as a condition of obtaining the patented invention.⁶⁹⁰ In developing the concept of patent misuse, the *Morton Salt* court, as an equity court, primarily was concerned with the public policy⁶⁹¹ that is carried out by the grant to an inventor of the special privilege of a patent monopoly.⁶⁹² The public policy behind patent law justifies exclusive rights in inventions within the granted monopoly but excludes from it all that is not embraced in the invention, and it "equally forbids the use of the patent to secure an exclusive right or limited monopoly not granted by the Patent Office and which it is contrary to public policy to grant."⁶⁹³ Whereas the latter component of the public policy, certainly including tie-in licenses, falls likewise in the ambit of competition law that seeks to curb anti-competitive conduct and promote a competitive marketplace, the first component of the policy underlying patents seeks

⁶⁸⁶ Neal Hartzog, *Gaining Momentum: A Review of Recent Developments Surrounding the Expansion of the Copyright Misuse Doctrine and Analysis of the Doctrine in its Current Form*, 10 MICH. TELECOMM. TECH. L. REV. 373, 405-6 (2004).

⁶⁸⁷ *Id.*, at 378-9.

⁶⁸⁸ See *Motion Picture Patents Co. v. Universal Film Mfg. Co.*, 243 U.S. 502 (1917).

⁶⁸⁹ *Morton Salt Co. v. G.S. Suppiger Co.*, 314 U.S. 488 (1942) [hereinafter *Morton Salt*].

⁶⁹⁰ *Morton Salt*, 314 U.S. at 490.

⁶⁹¹ 314 U.S. at 492 ("It is a principle of general application that...especially courts of equity, may appropriately withhold their aid where the plaintiff is using the right asserted contrary to the public interest.")

⁶⁹² *Id.*, at 494 (reasoning that the public policy behind the patent system were to draw a delicate balance between the social benefits of improved progress in "Science and the useful Arts" and the social costs of granting a "limited monopoly").

⁶⁹³ *Id.*, at 492.

to promote invention by granting a patent holder the ability to limit competition, and thereby goes beyond the domain covered by competition law.⁶⁹⁴

1.2.2.2. Post-Morton Salt Developments

Under *Morton Salt*, the original basis of the patent misuse defense was founded in equity and did not mirror the statutory competition laws. The Supreme Court distinguished the patent misuse defense from traditional antitrust analysis under the Clayton Act and did not draw a strong parallel to statutory competition law, neither did the *Morton Salt* court engage in lengthy analysis of the relevant markets, Morton Salt's share of it, the amount of salt tablets engrossed nor the anti-competitive impact of the restriction. In other words, where the patent holder is attempting to extend the reach of the patent monopoly, proof of anti-competitive effects is not required in order to render a patent unenforceable. However, the *Morton Salt* rule that patent misuse can occur in the absence of any antitrust violation as a matter of equity, was not upheld consistently by the circuit courts in the years following *Morton Salt*.⁶⁹⁵ As in the realm of patent law, the exclusive right⁶⁹⁶ allows patentees to act as temporary, limited monopolists, controlling the disposition of their protected goods in ways that might otherwise violate the competition laws. As a result of the inextricable interrelation between intellectual property law in general and competition law,⁶⁹⁷ courts have found patent misuse most often with tying arrangements or other anticompetitive activity that coincide with the types of activities that antitrust laws are designed to regulate.⁶⁹⁸ Hence, case law by itself, in essence beginning with *Morton Salt*, has spawned the patent

⁶⁹⁴ *Id.*, at 491-2 (ruling that the company was extending the scope of its patent monopoly beyond the limits allowed by the U.S. Constitution, the Patent Act, and public policy interests).

⁶⁹⁵ See Aaron Xavier Fellmeth, *Copyright Misuse and the Limitations of the Intellectual Property Monopoly*, 6 J. INTEL. PROP. L. 1, 14 (1998).

⁶⁹⁶ Patents confer an exclusive right to make, use, copy, sell, or import the respective subject of the patent for a limited time. See 35 U.S.C. § 154(a)(1) (2002).

⁶⁹⁷ See generally Fellmeth, *supra* note 695, at 2-5 (identifying the state of permanent tension between intellectual property and antitrust as that, on the one hand, the Constitution empowers Congress “[t]o protect the Progress of Science and useful Arts, by securing for limited Times to Authors and Investors the exclusive Right to their respective Writings and Discoveries.” U.S. CONST. Art.1, § 8, cl. 8 and, on the other hand, section 2 of the Sherman Act inhibits monopolizing any part of interstate trade, without mentioning any exception for exclusive intellectual property rights.).

⁶⁹⁸ Hartzog, *supra* note 686, at 401.

misuse defense in an attempt to prevent abuse of the federal government-granted monopoly on the subject of the patent.

1.2.2.3. Legislative Reflections in Patent Law

In the early 1970s, the Antitrust Division of the U.S. Department of Justice promulgated a list of licensing practices, known as the “Nine No-Nos”, that it presumed to be *per se* violations of the antitrust laws.⁶⁹⁹ The “Nine No-Nos”, including tying arrangements, appeared to be consistent with the misuse doctrine announced in *Morton Salt*.⁷⁰⁰ Each “No-No”-rule inherently constituted an attempt to extend the patent monopoly and thus qualified as patent misuse *per se*, whereby rendering the subject patents unenforceable. Since the early 1980s, the Justice Department was beginning to repudiate its list of *per se* illegal practices, and instead adopted a rule-of-reason approach, because of its emerging belief that the practices covered by the “Nine No-Nos” are frequently pro-competitive.⁷⁰¹ The thrust of the rule-of-reason standard is that, after finding that a patent owner has crossed the market power threshold, a court still may find that a tying arrangement does not constitute a misuse if significant pro-competitive potential of the tie-in outweighs its anticompetitive aspects.⁷⁰²

⁶⁹⁹ The “Nine No-Nos” were:

1. Requiring a licensee to purchase unpatented materials from the licensor (tying).
2. Requiring a licensee to assign to the licensor patents issued to the licensee after the licensing arrangement is executed.
3. Restricting a purchaser of a patented product in the resale of that product.
4. Restricting a licensee’s freedom to deal in products or services outside the scope of the patent.
5. Agreeing with a licensee that the licensor will not, without the licensee’s consent, grant further licenses to any other person.
6. Requiring that the licensee accept a “package” license.
7. Requiring royalties not reasonably related to the licensee’s sales of products covered by the patent.
8. Restricting the licensee’s sales of (unpatented) goods made with the licensed patented process.
9. Requiring a licensee to adhere to specified or minimum prices in the sale of the licensed products.

See Jere M. Webb & Lawrence A. Locke, *Intellectual Property Misuse: Developments in the Misuse Doctrine*, 4 HARV. J. L. & TECH. 257, 260-61 n.23 (1991).

⁷⁰⁰ Daniel P. Homiller, *Patent Misuse in Patent Pool Licensing: From National Harrow to “The Nine No-Nos” to not likely*, 2006 Duke L. & Tech. Rev. 7 (2006).

⁷⁰¹ See Webb & Locke, *supra* note 699, at 261 n.24.

⁷⁰² Regarding monopolies, the Department stated that “[m]arket power or even a monopoly that is the result of superior effort, acumen, foresight, or luck does not violate the antitrust laws [and] [t]he owner of intellectual property is entitled to enjoy whatever market power the property itself may confer.” *Antitrust Guidelines for International Operations*, reprinted in 55 Antitrust & Trade Reg. Rep. (BNA) No. 1391, at S-16 (Nov. 17, 1988) [hereinafter Antitrust Guidelines].

Due to the close relatedness of patent and antitrust, the changing policy underlying antitrust law affected the patent misuse legislation in late 1988.⁷⁰³ Section 271(d)(5) Patent Misuse Reform Act⁷⁰⁴ prohibits a finding of patent misuse of a tie-in license “unless, in view of the circumstances, the patent owner has market power in the relevant market for the patent or patented product on which the license or sale is conditioned.”⁷⁰⁵ The provision codifies a rule-of-reason standard to determine whether a tie-in case involves patent misuse.

1.2.3. Lasercomb and Its Progeny: The Copyright Misuse Doctrine

1.2.3.1. Lasercomb America, Inc. v. Reynolds (1990)

The Fourth Circuit⁷⁰⁶ was the first appellate court⁷⁰⁷ to expressly recognize and apply the doctrine of copyright misuse as an affirmative defense to infringement. In

⁷⁰³ As a result of an erroneous decision by the United States Court of Appeals for the Seventh Circuit in *Bendix Corp. v. Balax, Inc.*, 471 F.2d 149, 158 (7th Cir. 1972), in referring to an antitrust counterclaim as an affirmative patent misuse claim, the United States Senate passed a reaction-driven bill in 1988 that would have required a strict standard of antitrust violation for a claim of patent misuse to succeed. However, the attempt to clarify the matter by legislation was in vain since the measure was never adopted by the House of Representatives. See 6 DONALD S. CHISUM, CHISUM ON PATENTS § 19.04[1][f] (2005).

⁷⁰⁴ Patent Misuse Reform Act of 1988, Pub. L. No.100-703, 102 Stat. 4676 (1988) [hereinafter Patent Misuse Act].

⁷⁰⁵ 35 U.S.C. § 271(d) reads in relevant part:

“No patent owner otherwise entitled to relief for infringement or contributory infringement of a patent shall be denied relief or deemed guilty of misuse or illegal extension of the patent right by reason of his having done one or more of the following:...

(4) refused to license or use any rights to the patent; or

(5) conditioned the license of any rights to the patent or the sale of the patented product on the acquisition of a license to rights in another patent or purchase of a separate product, unless, in view of the circumstances, the patent owner has market power in the relevant market for the patent or patented product on which the license or sale is conditioned.”

⁷⁰⁶ For the legal status before *Lasercomb* see the Seventh Circuit approach adopted by Circuit Judge Posner in *Saturday Evening Post Co. v. Rumbleseat Press, Inc.*, 816 F.2d 1191 (7th Cir. 1987) (holding that the misuse doctrine is only targeted at conduct that violates the antitrust laws or at least that comes close to violating those laws and thus concluding that “[i]f misuse claims are not tested by conventional antitrust principles, by what principles shall they be tested? Our law is not rich in alternative concepts of monopolistic abuse; and it is rather late in the date to try to develop one without in the process subjecting the rights of patent holders to debilitating uncertainty. This point applies with even greater force to copyright misuse, where the danger of monopoly is less.” *Id.* at 1200. However, In spite of the disastrous consequences anticipated by Posner if the misuse doctrine is not tied to antitrust principles, three years later, the Fourth Circuit in *Lasercomb* made clear that misuse of copyrights encompasses more than just antitrust violations.

⁷⁰⁷ Only one court before *Lasercomb* had applied a copyright misuse defense to bar an infringement claim. See *M. Witmark & Sons v. Jensen*, 80 F. Supp. 843 (D. Minn. 1948).

Lasercomb America, Inc. v. Reynolds,⁷⁰⁸ plaintiff Lasercomb America, Inc. (“Lasercomb”) licensed its copyrighted software to the defendants. The software enabled steel rule die manufacturers to utilize a computer to create a design and then direct manufacturing systems.⁷⁰⁹ The objectionable provisions of Lasercomb’s standard licensing agreement prohibited licensees from independently creating or selling “computer assisted die making software for a period of one hundred years.”⁷¹⁰ The defendants, a competitor of Lasercomb, first copied the software for unauthorized private use, circumventing protective devices and avoiding additional fees, and then marketed its own software that was virtually a direct copy of Lasercomb’s program. The defendants asserted copyright misuse as an affirmative defense to the infringement action brought by Lasercomb. The Fourth Circuit found that the language of the licensing agreement improperly prohibited licensees’ use of their own ingenuity to create software implementing the idea expressed in Lasercomb’s software.

1.2.3.2. The Practice Management Decision (1997)

In *Practice Management Information Corporation. v. American Medical Association*,⁷¹¹ the Ninth Circuit—after *Lasercomb* the next circuit to fully adopt the copyright misuse doctrine—the American Medical Association (“AMA”) granted the Health Care Financing Administration (“HCFA”) a “nonexclusive, royalty free and irrevocable license to use, copy, publish and distribute” a coding system for medical procedures (“CPT”)⁷¹² in exchange for the HCFA agreeing not to use any other system than the CPT and its use in “programs administered by the HCFA, by its agents, and by other agencies whenever possible.”⁷¹³ By negotiating a contract in which HCFA agreed to use the CPT exclusively the Ninth Circuit held AMA’s anticompetitive licensing provision as the basis for misuse of

⁷⁰⁸ 911 F.2d 970 (4th Cir. 1990) [hereinafter *Lasercomb*].

⁷⁰⁹ *Lasercomb*, 911 F.2d at 971.

⁷¹⁰ 911 F.2d at 973.

⁷¹¹ 121 F.3d 516 (9th Cir. 1998), opinion amended, 133 F.3d 1140 [hereinafter *Practice Management*].

⁷¹² The AMA designed a coding system to identify medical procedures and that system was published in a guide called the Physician’s Current Procedural Terminology (“the CPT”).

⁷¹³ *Practice Management*, 121 F.3d at 517-8.

copyright⁷¹⁴—without delving into a substantive analysis of the public policy concept of copyright⁷¹⁵ Despite of the court’s antitrust-like analysis,⁷¹⁶ and rather following *Lasercomb*, the court reiterated that a finding of copyright misuse does not require proof of an antitrust violation.

1.2.3.3. The Alcatel decision (1999)

*Alcatel USA, Inc. v. DGI Technologies, Inc.*⁷¹⁷ is another example for a circuit court treating the defense of copyright misuse as a question of a strict antitrust violation and thus may be quoted as an outgrowth of antitrust-based copyright misuse in its uttermost extent.⁷¹⁸ Alcatel USA, Inc. (“Alcatel”, formerly known as DSC Communications Corp.) sold unpatented telephone switches along with a license agreement directing its customers, in essence, not to use Alcatel’s copyrighted operating system software in conjunction with any non-Alcatel microprocessor cards.⁷¹⁹ A conflict arose when DGI Technologies, Inc. (“DGI”), a manufacturer of telephone switch expansion cards, sought to establish a foothold at the market in Alcatel-competing microprocessor cards capable of functioning along with or in lieu of Alcatel’s own cards. In order to market competing microprocessor cards, DGI lawfully obtained one of Alcatel’s microprocessor cards and by decompiling

⁷¹⁴ The Ninth Circuit stated that “[t]he adverse affects of the licensing agreement” were apparent and “gave the AMA a substantial and unfair advantage over its competitors” in the marketplace. *See* 121 F.3d at 521.

⁷¹⁵ Kathryn Judge, *Rethinking Copyright Misuse*, 57 STAN. L. REV. 901, 922 (2004) (criticizing the Ninth Circuit treating the provision as a violation *per se* and so not to consider its potential welfare-enhancing effects).

⁷¹⁶ Despite the fact that the circuit court dismissed various arguments made by the AMA that likely would be considered in a rule-of-reason antitrust analysis (e.g., consideration of AMA’s market power), the *Practice Management* court arguably relied on anticompetitive concerns which have eventually given rise to its misuse holding. *See id.* at 921.

⁷¹⁷ *DSC Communications Corp. v. DGI Technologies, Inc.*, 81 F.3d 597 (5th Cir. 1996) (while adopting the doctrine of copyright misuse in 1996, the circuit court overruled the judge’s discretion and reinstated the jury finding of copyright misuse in *Alcatel USA, Inc. v. DGI Technologies, Inc.*, 166 F.3d 772 (5th Cir. 1999), *reh’g and reh’g en banc denied*, 180 F.3d 267 [hereinafter Alcatel]).

⁷¹⁸ *See also In Re Independent Service Organizations Antitrust Litigation*, 964 F.Supp. 1469, 1477 (D. Kan. 1997) (stating that an alleged infringer can establish defense of copyright misuse by showing either that defendant violated antitrust laws, or that defendant illegally extended its monopoly beyond the scope of copyright or violated public policies underlying copyright laws).

⁷¹⁹ The relevant terms of the software license provided that (1) the operating system software remains the property of DSC; (2) the customer has the right to use the software only to operate its switch; (3) the customer is prohibited from copying the software or disclosing it to third parties; and (4) the customers are authorized to use the software only in conjunction with DSC-manufactured equipment. *See Alcatel*, 166 F.3d at 777.

Alcatel's software was able to develop a similar microprocessor card, the competitive purpose of which would enable the card to accept a download of Alcatel's software upon activation in the switch system.⁷²⁰ By this way, DGI was enabled to produce expansion cards compatible with the controlling operating system to use with Alcatel switches. In response to Alcatel's infringement suit DGI claimed that Alcatel had misused its copyright by virtue of Alcatel's license agreement according to which the use of Alcatel's operating system software was restricted to Alcatel-manufactured hardware, thereby seeking to obtain patent-like protection of its hardware—its microprocessor card—through the enforcement of copyright granted in its operating system software.⁷²¹

When it comes to the standard applied in *Alcatel*—whether public policy-based or antitrust-like—the facts of *Alcatel* differ from *Lasercomb* in a decisive nuance, in that the licensing agreement did not prevent independent development of Alcatel's software and thereby the independent development of a competing product, but rather did it restrict unauthorized *access* to the ideas expressed.⁷²² Thus, copyright misuse in *Lasercomb* stemmed from the copyright holder's attempt to suppress competition by way of exercising his copyright, whereas in *Alcatel* copyright misuse was found in the use of “copyrights to indirectly gain commercial control over products Alcatel d[id] not have copyrighted.”⁷²³ In establishing a copyright misuse standard, the *Alcatel* court struck a new path by looking beyond the plain wording of the license agreement and questioning whether there existed a relevant market for expansion products, such as DGI's cards.⁷²⁴ The court found that DGI was effectively prevented from developing its product, thereby securing for Alcatel a *de facto* monopoly on its uncopyrighted and unpatented microprocessor cards.⁷²⁵

1.2.2.4. The Napster Decision (2002)

⁷²⁰ *Id.*

⁷²¹ *Id.* at 793.

⁷²² Theodore Dorenkamp, *Copyright Misuse or A Right To Compete?: A Critique of Alcatel USA v. DGI Technologies*, 9 TEX. INTELL. PROP. L.J. 269, 299-300 (2001).

⁷²³ *Alcatel*, 166 F.3d at 793.

⁷²⁴ Judge, *supra* note 715, at 919.

⁷²⁵ 166 F.3d at 793-4.

In *In Re Napster, Inc. Copyright Litigation*,⁷²⁶ five major recording companies filed a complaint for copyright infringement against Napster, an Internet service that facilitated the downloading of MP3 music files and thereby committed the unauthorized digital distribution and reproduction of the plaintiffs' copyrighted music.⁷²⁷ In 2001, the plaintiffs formed two joint ventures to provide platforms for the digital distribution of music—namely, MusicNet, a venture between EMI, BMG, and Warner, and Pressplay with Sony and Universal as members.⁷²⁸ In the same year, Napster entered into a license agreement with MusicNet to distribute the music from the catalogs of the three participating MusicNet plaintiffs and any other label that licenses its catalog to MusicNet.

Although on its face the license was non-exclusive, meaning Napster was free to obtain licenses from any of the record company plaintiffs, in fact the license well was of an exclusive nature as MusicNet has the ultimate control over which music Napster licenses as Napster was required to go through MusicNet,⁷²⁹ i.e., first MusicNet had to approve to license that music in order for Napster to lawfully have access to it. In other words, Napster could either sign the overly restrictive license agreement to gain access to the catalogs of major record companies, or refuse to sign the agreement and have virtually no *access* to most commercially available music.⁷³⁰ The court acknowledged that although antitrust violations could also qualify as a misuse of copyright, Napster must prove more than a general antitrust violation.⁷³¹ It must “establish a nexus between...alleged anti-competitive actions

⁷²⁶ Napster, 191 F.Supp.2d 1087 (N.D. Cal. 2002) [hereinafter Napster].

⁷²⁷ A&M Records, Inc. v. Napster, Inc., 239 F.3d 1004 (9th Cir. 2001).

⁷²⁸ Napster, 191 F.Supp.2d at 1105.

⁷²⁹ Napster attempted to invoke the copyright misuse defense against the plaintiffs resting on two bases: first, it contended that the license agreement with MusicNet was unduly restrictive, and, second, it asserted that even if that particular agreement was not unduly restrictive, plaintiffs' practices as they enter the market for the digital distribution of music were sufficient anti-competitive as to give rise to a misuse defense. Napster referred to section 19.1 of the license agreement, according to which Napster was prevented from entering into any license agreement with any individual plaintiffs until March 1, 2002—the so-called “Initial Exclusivity Period”. By the same agreement, MusicNet would be entitled to terminate the agreement on ninety-days' notice if Napster entered into any individual license with any of the plaintiffs, including the MusicNet plaintiffs, after March 1, 2002. Further, a pricing structure under which Napster would be charged higher fees if it failed to use MusicNet as its exclusive licensor for content was laid out in the agreement in question. 191 F.Supp.2d at 1105-6.

⁷³⁰ *Id.*, at 1107.

⁷³¹ *Id.*, at 1108.

and [plaintiffs'] power over copyrighted material.”⁷³² *In casu*, the court refused to grant the plaintiffs’ motion for summary judgment based upon the possibility that the plaintiffs engaged in price fixing, an action that “carries antitrust and public policy considerations that may be relevant to misuse.”⁷³³

1.2.2.5. The Video Pipeline Decision (2003)

The Third Circuit in *Video Pipeline, Inc. v. Buena Vista Home Entertainment, Inc.*⁷³⁴ adopted the copyright misuse defense and strongly supported the policy approach while, at the same time, strictly forbearing from undergoing an antitrust-based analysis.⁷³⁵ Video Pipeline, Inc. (“Video Pipeline”) used copyrighted motion pictures to make short trailers, which were then made available for money to the Video Pipeline’s clients, which were video retailers, for the viewing by retail customers on the retailers’ Internet websites, for the purpose of promoting sales of the copyrighted videos. According to the license agreement Video Pipeline as the licensee was prohibited from criticizing the copyright owner, its works, or the entertainment industry generally.⁷³⁶ Although the agreement sought to restrict expression, it would not likely interfere with creative expression to such a degree that it significantly affected the policy interest in increasing the public store of creative activity.⁷³⁷ The court thus found no indication that the owner of the copyright was using its copyright to restrain competition in the video retail market, and so held that the copyright owner was not subject to the copyright misuse defense.⁷³⁸ The court stated that the purpose and value of copyrights was to

⁷³² *Id.*, (citations omitted).

⁷³³ *Id.*, at 1109.

⁷³⁴ 342 F.3d 191 (3rd Cir. 2003), *cert. denied*, 540 U.S. 1178 (2004) [hereinafter Video Pipeline].

⁷³⁵ Video Pipeline, 342 F.3d at 204.

⁷³⁶ 342 F.3d at 203.

⁷³⁷ The court rejected the misuse defense given the facts that Disney, the licensor, did not attempt to restrict its licensees from expressing critical views elsewhere on the Internet or outside the cyberspace altogether. *Id.* at 206.

⁷³⁸ Emphasis was clearly laid on determining whether the license agreements in question undermined the underlying public policy rationale for the misuse doctrine. The Third Circuit held that “[the copyright misuse doctrine] might operate beyond its traditional anti-competition context.” *See id.* at 204.

promote the creation and publication of free expression by stimulating artistic creativity for the general public welfare.⁷³⁹

1.2.3.6. The WIREdata Decision (2003)

In 2003, also the Seventh Circuit⁷⁴⁰ tackled the copyright misuse issue marginally in *Assessment Technologies of Wisconsin L.L.C. v. WIREdata, Inc.*⁷⁴¹ The plaintiff, Assessment Technologies of Wisconsin (“AT”), was the owner of a copyrighted software designed to compile and store real estate tax assessment data. AT licensed its software to municipalities in Wisconsin to collect data. The defendant company, WIREdata, Inc. (“WIREdata”), sought public information about a number of properties in Wisconsin from the municipalities in which the properties were located. However, three municipalities, which used software copyrighted by Assessment Technologies (“AT”) to compile the relevant data, refused to provide the requested information for fear that doing so would infringe AT’s copyright. AT initiated the federal suit, claiming that the release would violate its copyright, because the data could not be extracted without infringement of its copyright.⁷⁴² The Seventh Circuit held the copyright owner to block WIREdata from obtaining non-copyrightable data in the public domain.⁷⁴³ *In dictum*,⁷⁴⁴ Circuit Judge Posner articulated that AT’s business practices “might constitute copyright misuse,”⁷⁴⁵ especially when, as was the case here, the complete data were unavailable

⁷³⁹ If Disney were required to license its trailers for display on any web site and regardless of the content displayed with its copyrighted works, Disney would be likely to refuse to license at all any online display of its works on web sites. *Id.* at 206.

⁷⁴⁰ The Seventh Circuit seems to concede to the absolution of an antitrust-oriented approach in copyright misuse context declared in *In re Napster*. See *WIREdata*, 350 F.3d 640 at 647. *But see Saturday Evening Post Co. v. Rumbleseat Press, Inc.*, 816 F.2d 1191, 1200 (7th Cir.1987) (Circuit Judge Posner extending his view of patent misuse to copyright misuse, characterizing the doctrine as a defense that should be evaluated under traditional antitrust principles).

⁷⁴¹ 350 F.3d 640 (7th Cir. 2003) [hereinafter *WIREdata*].

⁷⁴² *WIREdata*, 350 F.3d at 641-2.

⁷⁴³ As opposed to the compilation itself, the mere data at issue were facts which are not subject to copyright protection. *Id.*

⁷⁴⁴ Posner, writing for a unanimous opinion in *WIREdata* speculated, but did not rule on, the issue of copyright misuse.

⁷⁴⁵ *De facto* the license agreement would result in forbidding municipalities licensed by AT to share the data in their tax-assessment databases with each other even for the purpose of comparing or coordinating their assessment methods.

anywhere else.⁷⁴⁶ Posner expressly relegated the antitrust approach⁷⁴⁷ and instead made the copyright owner's motives subject to the misuse analysis in a hitherto unprecedented way:

[F]or a copyright owner to use an infringement suit to obtain property protection, here in data, that copyright law clearly does not confer, hoping to force a settlement or even achieve an outright victory over an opponent . . . is an abuse of process.⁷⁴⁸

The Seventh Circuit bypassed the public policy rationale and, for the first time, considered copyright misuse on ground of abuse of process.

1.2.4. Two Standards—One Copyright Misuse Doctrine

To date, courts facing copyright misuse defenses have either invoked the underlying public policy of the copyright misuse in favoring dissemination of ideas and expression of information, or preserving of the courts' integrity, or/and ruled in favor of economic competition.⁷⁴⁹ The question judges thus face is whether they shall analyze copyright misuse by way of focusing on either antitrust standards or public policy considerations. The answers are far from certain. Judges avail themselves of one of the following two standards in adjudicating misuse defense.

1.2.4.1. The Public Policy Rationale

In *Morton Salt*, the Supreme Court sidestepped the antitrust inquiry⁷⁵⁰ and ruled instead that misuse encompasses practices that are “contrary to public policy” or that have an “adverse effect upon the public interest.”⁷⁵¹ In analogizing the public policy approach of *Morton Salt*, copyright misuse occurs when the copyright owner

⁷⁴⁶ *WIREDATA*, 350 F.3d at 646-47.

⁷⁴⁷ *Id.* (arguing that the antitrust approach would not have been responsive to the abuse of process in *WIREDATA* and, moreover, noting that copyright misuse confined to antitrust would render the doctrine redundant).

⁷⁴⁸ *Id.* at 647.

⁷⁴⁹ Note, *Clarifying the Copyright Misuse Defense: The Role of Antitrust Standards and First Amendment Values*, 104 HARV. L. REV. 1289, 1308 (1991).

⁷⁵⁰ *Morton Salt*, 314 U.S. at 490, 493-94; see Fellmeth, *supra* note 695, at 14 (referring to the Supreme Court holding the contract limitation unenforceable and contrary to public policy, even though no antitrust violation was alleged).

⁷⁵¹ 314 U.S. at 494.

expands the statutory copyright monopoly in order to gain control over areas outside the scope of the monopoly.⁷⁵² The relevant public policy is anchored in the Intellectual Property Clause of the Constitution of the United States,⁷⁵³ according to which the privilege of a patent monopoly, which includes inventions within the monopoly granted to the patentee but similarly excludes from it all that is not embraced in the invention, serves “to promote the Progress of Science and useful Arts.”⁷⁵⁴ Since *Morton Salt* is regarded as *de facto* genesis of the copyright misuse doctrine, the same scope-of-the-grant approach was adopted by the Fourth Circuit in *Lasercomb*.⁷⁵⁵ The court found that Lasercomb’s standard license agreement, which barred licensees from developing their own competing software, extended Lasercomb’s copyright protection beyond the program’s expression in an effort to encompass also its underlying unprotected idea or function.⁷⁵⁶ As a result of *Lasercomb*, misuse can be regarded to exist whenever a copyright holder attempts to extend the scope of his limited exclusive rights beyond the intended scope of the copyright grant.

As for the time being, copyright policy has appeared in different facets in different cases. Against the backdrop of the copyright term under the Copyright Act,⁷⁵⁷ the duration of the non-compete term (one hundred years by license⁷⁵⁸) in *Lasercomb* had the potential to run longer than the copyright itself. The license clause could be interpreted as Lasercomb’s attempt to stipulate for itself a right beyond those granted to it by the Copyright Act in order to prevent others from competing in the field.⁷⁵⁹ In *Alcatel*,⁷⁶⁰ the Fifth Circuit held that Alcatel’s action in preventing card manufacturing competitors from developing products that were compatible with Alcatel’s copyrighted software was regarded as a copyright owner exceeding the

⁷⁵² *Napster*, 191 F.Supp.2d at 1103.

⁷⁵³ U.S. CONST. art. I, § 8, cl. 8. (providing that “[The Congress shall have Power] To promote the Progress of Science and useful Arts, by securing for Limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries”).

⁷⁵⁴ *Morton Salt*, 314 U.S. at 492.

⁷⁵⁵ *Lasercomb*, 911 F.2d at 978 (stating that the principle issue in determining whether a copyright owner has misused its copyright is whether the copyright owner has expanded the copyright use beyond its scope to subvert public policy).

⁷⁵⁶ 911 F.2d at 979.

⁷⁵⁷ 17 U.S.C. § 302(b) (providing for a copyright term encompassing lifetime of the author plus 70, respectively, almost certainly 95 years (works made for hire) years *post mortem auctoris*).

⁷⁵⁸ *Lasercomb*, 911 F.2d at 973.

⁷⁵⁹ Judge, *supra* note 715, at 917.

⁷⁶⁰ *Alcatel*, 166 F.3d 772.

scope of the grant in the copyrightable expression contained in the software by way of gaining a monopoly in an uncopyrightable item (i.e. microprocessor cards) and was therefore subject to misuse of copyright.⁷⁶¹ Several years later, the *Video Pipeline* court recognized the “public policy” as inherent in the idea/expression dichotomy of the copyright concept. The Third Circuit acknowledged⁷⁶² that the idea/expression dichotomy as applied in the copyright misuse defense constitutes a necessary complement to promoting the creation and publication of free expression in so far as it serves the public interest in the dissemination of ideas.⁷⁶³ The court in *Video Pipeline* reflected on an even broader guise of the misuse defense “that a copyright holder could leverage its copyright to restrain the creative expression of another without engaging in anti-competitive behavior or implicating the fair use and idea/expression doctrines.”⁷⁶⁴

The court in *Lasercomb* clearly refrained from applying an antitrust standard in holding that Lasercomb misused its copyright by paying particular attention to the language of the license agreement.⁷⁶⁵ Had the antitrust approach been applied to *Lasercomb*, the question would have been whether it was the anti-competitive nature of the non-compete provision that would give rise to the misuse⁷⁶⁶ even “[without necessarily being] a violation of antitrust law in order to comprise an equitable defense to an infringement action.”⁷⁶⁷ Given the facts in *Lasercomb*, the court would have had to focus on the actual effects on competition or market power of the plaintiff.

1.2.4.2. The Antitrust-Like Standard

“[T]he aims and objectives of [intellectual property] and antitrust laws may seem, at first glance, wholly at odds. However, the two bodies of law are actually

⁷⁶¹ Hartzog, *supra* note 686, at 386-87.

⁷⁶² *Id.*, at 205 n.13.

⁷⁶³ Note, *supra* note 749, at 1305-06. In other words, it is the non-copyrightability of ideas that leaves them in the public domain, thus being accessible to the general (artistic) public to “promote the Progress of Science and useful Arts”. *Video Pipeline*, 342 F.3d at 204 (citing U.S. CONST. art. I, § 8, cl. 8.).

⁷⁶⁴ *Id.* at 205.

⁷⁶⁵ Frischmann & Moylan, *supra* note 684, at 890.

⁷⁶⁶ Judge, *supra* note 715, at 917.

⁷⁶⁷ *Lasercomb*, 911 F.2d at 978.

complementary, as both are aimed at encouraging innovation, industry and competition.”⁷⁶⁸ While intellectual property laws provide incentives for innovation and its dissemination and commercialization by establishing enforceable property rights for the creators of new and useful products, more efficient processes, and original works of expression, competition laws promote the goal of innovation and enhancing consumer welfare by prohibiting certain actions that may harm competition with respect to either existing or new ways of serving consumers.⁷⁶⁹ The absence of intellectual property rights would entail rapid exploitation of the efforts of innovators and investors without compensation. Rapid imitation would reduce the commercial value of innovation and erode incentives to invest, ultimately to the detriment of consumers.⁷⁷⁰ Both bodies of law pursue the same goal⁷⁷¹ but are independent from each other in the way how to achieve that goal.⁷⁷² Courts are challenged to walk the thin line between competition law and intellectual property rights. In doing so they are advised to apply the fact-based rule-of-reason approach on a case-by-case basis. To put it in words used by the United States Department of Justice and the Federal Trade Commission:

In the vast majority of cases, restraints in intellectual property licensing arrangements are evaluated under the rule of reason. The [United States Department of Justice and the Federal Trade Commission’s] general approach in analyzing a licensing restraint under the rule of reason is to inquire whether the restraint is likely to have anticompetitive effects and, if so, whether the restraint is reasonably necessary to achieve procompetitive benefits that outweigh those anticompetitive effects. ... Application of the rule of reason generally requires a comprehensive inquiry into market conditions.⁷⁷³

⁷⁶⁸ Atari, 897 F.2d at 1576.

⁷⁶⁹ United States Department of Justice and the Federal Trade Commission, Antitrust Guidelines for the Licensing of Intellectual Property (April 6, 1995), *available at* <http://www.usdoj.gov/atr/public/guidelines/0558.htm> (last visited Nov. 10, 2006) [hereinafter IP Guidelines].

⁷⁷⁰ *See id.*

⁷⁷¹ *But see Alfred Bell & Co. v. Catalda Fine Arts, Inc.*, 191 F.2d 99, 106 (2d Cir. 1951) (recognizing a conflict of policies between antitrust principles and copyright monopoly which necessitates a balance between enforcing the antitrust laws and that of preventing piracy of copyrighted material).

⁷⁷² Hartzog, *supra* note 686, at 400-1.

⁷⁷³ *See* IP Guidelines, *supra* note 769.

Courts will face questions like whether a copyright owner may exclude others from using his copyrighted technical know-how, or may refuse to license the use of his technology at all despite any immediate harm to consumers. The judges' answers are rich in diversity.⁷⁷⁴ The anti-competitive-like approach has frequently been held a dispositive factor in cases involving copyright misuse,⁷⁷⁵ even though the tainted conduct has not met—and has actually never been demanded to meet—the level of a traditional antitrust violation in intellectual property cases.⁷⁷⁶ Proponents of antitrust-based copyright misuse refer in their misuse analysis to patent misuse in its modern day application,⁷⁷⁷ because what the courts clearly have established is a general patent misuse doctrine parallel to antitrust claims.⁷⁷⁸ Legal certainty and transparency are claimed to be associated with this straight analysis as undergone in patent cases and therefore claimed to be adopted in copyright cases likewise.⁷⁷⁹ In accordance with the modern day application of competition law, we can distinguish misuse *per se* and the rule-of-reason test which applies to all other activities not covered under the *per se* rule.⁷⁸⁰

The *per se* rule is defined as the judicial principal in competition law that a trade

⁷⁷⁴ See cases cited *supra* in Parts VII.1.2.3. et seq.

⁷⁷⁵ In particular, the Seventh Circuit had formerly insisted on applying a strict antitrust-modeled misuse analysis in copyright cases. Also, the rhetorical question which, if any, other standard should be applied in copyright misuse cases, given a lack of equally tested or applicable alternatives, tips in favor of a principled antitrust approach.

See *USM Corp. v. SPS Technologies, Inc.*, 694 F.2d 505, 512 (7th Cir.1982), and reiterated in *Saturday Evening Post Co. v. Rumbleseat Press, Inc.*, 816 F.2d 1191, 1200 (7th Cir. 1987) (“If misuse claims are not tested by conventional antitrust principles, by what principles shall they be tested? Our law is not rich in alternative concepts of monopolistic abuse; and it is rather late in the date to try to develop one without in the process subjecting the rights of patent holders to debilitating uncertainty.”); *but see* Fellmeth, *supra* note 695, at 36 (criticizing the question to assume a false dichotomy and reformulating it as to ask how courts can tailor the legal rule in cases such as this one to be consistent with the purpose of the intellectual property law as an exception to the antitrust laws).

⁷⁷⁶ See generally Hartzog, *supra* note 686, at 400 (identifying circuit courts claiming by analogy to patent misuse and its (statutory) development that copyright misuse should be found only either when an antitrust violation occurs (so-called “strict” antitrust approach), or, at least, when the alleged misuse affirmatively has undergone the analysis according to antitrust principles); Judge, *supra* note 715, at 924; *but see* Fellmeth, *supra* note 695, at 10-1 (pointing out the unsteady acceptance of the antitrust approach).

⁷⁷⁷ Frischmann & Moylan, *supra* note 684, at 898.

⁷⁷⁸ Fellmeth, *supra* note 695, at 15.

⁷⁷⁹ *But see id.* at 18-19 (distinguishing between *per se* patent misuse as a bright line rule which is susceptible to being criticized as too rigid and rule-of-reason with its complicating factor that the question is not only whether the allegedly abusive practice is anti-competitive, but whether the practice is anti-competitive outside the scope of the patent).

⁷⁸⁰ See generally Frischmann & Moylan, *supra* note 684, at 898; Fellmeth, *supra* note 695, at 16; Hartzog, *supra* note 686, at 400.

practice violates the Sherman Act if the practice is a restraint of trade,⁷⁸¹ regardless of whether it actually harms anyone because an anti-competitive effect is presumed if the rightholder engages in a generally forbidden practice. Tying-in agreements are a classic example covered by *per se* misuse. Put in a copyright context, such tying-in arrangements encompass practices conditioning the sale or license of copyrighted goods (usually the desirable computer software) on the purchase of either copyrighted or uncopyrighted goods (often undesirable hardware).⁷⁸²

The rule-of-reason constitutes the second judicial standard applied in competition law. Under the rule, an intellectual property owner's trade practice violates the Sherman Act only if the practice is an unreasonable restraint of trade, based on economic factors.⁷⁸³ Whereas a *per se* analysis generally tends to preclude any defense to the alleged violation and tailors the presentation of evidence to what is needed to characterize the contested conduct as fitting within the *per se* classification, the rule-of-reason analysis involves a comprehensive, fact-intensive inquiry in which courts assess the reasonableness of alleged conduct.⁷⁸⁴

In a copyright infringement context, courts, first, must investigate whether the restraint is within the scope of the copyright monopoly. At this level courts have to engage in evaluating the scope of the protection granted to copyright holders; i.e., in essence, to look at the alleged misuse activity from a public policy perspective.⁷⁸⁵ If the restraint is within the scope of the copyright granted, then the activity is *per se* illegal. If not, however, courts have to determine further whether the questionable conduct on the whole promotes or restricts competition. This part of the test requires courts to balance the anti- and pro-competitive effects of a particular asserted misuse activity.⁷⁸⁶ The second part of the rule-of-reason test is necessary as intellectual property generally is assumed to imply pro-competitive benefits and not to confer market power in an antitrust context upon its owner, and

⁷⁸¹ BLACK'S LAW DICTIONARY 1178 (8th ed. 2004).

⁷⁸² Note, *supra* note 749, at 1302-03; Fellmeth, *supra* note 695, at 24.

⁷⁸³ BLACK'S LAW DICTIONARY 1360 (8th ed. 2004).

⁷⁸⁴ Frischmann & Moylan, *supra* note 684, at 878.

⁷⁸⁵ Hartzog, *supra* note 686, at 402.

⁷⁸⁶ Frischmann & Moylan, *supra* note 684, at 898; Hartzog, *supra* note 686, at 400 (requiring the "net competitive effects" of misuse to subject to a close scrutiny).

that any market power (or even monopoly) established through intellectual property rights is valid.⁷⁸⁷

1.2.5. Procedural Nature of the Misuse Defense

One must not lose sight of misuse being an affirmative defense to an infringement claim and not, unlike an antitrust violation, a cause of action. As result thereof, a copyright holder who misuses his copyright is unable to enforce that copyright against an infringer for the duration of the misuse, even if the infringer suffered no actual harm from the misuse. However, the flipside of the coin is that an infringer cannot file a suit against the misuser unless the latter commits an independent antitrust violation.⁷⁸⁸

The role of competition law must not be overestimated, standing alone, offers full protection of the public interest. In addition to the public policy goal of preventing anti-competitive business practices, society has also a fundamental interest in the free flow of ideas and information. Although most of the copyright cases involve unduly restrictive licensing schemes,⁷⁸⁹ the public interest may be harmed even if the infringing plaintiff's conduct does not threaten to undermine competitive conditions.⁷⁹⁰ As, for example, in *Lasercomb*,⁷⁹¹ license restrictions often intend to remove ideas from the public domain even though they do not enable licensors to restrict economic competition or secure monopoly power. The public injury results not from the departure of market price from competitive levels but from the suppression of the free exchange of facts and ideas guaranteed by the limited principles of copyright law.⁷⁹²

1.2.6. Current Impact of the Intellectual Property Misuse Doctrines

⁷⁸⁷ See IP Guidelines, *supra* note 769.

⁷⁸⁸ Fellmeth, *supra* note 695, at 10-11.

⁷⁸⁹ See, e.g., *Practice Management*, 121 F.3d 516 (finding licensing agreement precluding use of competitor's products to be misuse); *Alcatel*, 166 F.3d 772 (overturning district court's rejection of misuse defense, based on licensing agreement allowing plaintiff to gain control over uncopyrighted products); *Lasercomb*, 911 F.2d 970 (concluding that exclusive licensing clause inhibiting licensees from developing own products constituted misuse).

⁷⁹⁰ Note, *supra* note 749, at 1303.

⁷⁹¹ *Lasercomb*, 911 F.2d 970 at 973, 978.

⁷⁹² Note, *supra* note 749, at 1303.

1.2.6.1. Patent Misuse Doctrine

The Patent Misuse Act in its section 271(d) confines misuse to the category uniformly considered most problematic: tying-in arrangements. However, such agreements are condemned only upon proof of market power held by the patentee. In other words, even otherwise illegal tying does not constitute a defense to infringement if the patentee lacks market power in the relevant market.⁷⁹³ Beyond the Patent Misuse Act, a patent owner is left with two ways of how to exploit a patent in an improper manner, to wit, either by violating the federal antitrust laws or, as in *Morton Salt*, by extending the patent beyond its lawful scope while falling short of an antitrust violation. One court summarized the doctrine of patent misuse to the point as “[a] concept [that] arose to restrain practices that did not in themselves violate any law, but that drew anticompetitive strength from the patent right, and thus were deemed to be contrary to public policy.”⁷⁹⁴ Since the Patent Misuse Act does not make any contribution to illuminate the shape and impact of the patent misuse beyond this expressly enlisted form of misuse⁷⁹⁵ the legal outcomes have still remained far from certain when antitrust claims arise that implicate patent issues.

In the recent decision *Illinois Tool Works, Inc. v. Independent Ink, Inc.*,⁷⁹⁶ the Supreme Court decided, in light of the 1988 amendments to the patent laws, to overrule the rebuttable presumption that patentees possess market power when they tie-in the purchase of unpatented goods with the purchase of the patented product. The Supreme Court held that patents do not confer *per se* market power for purposes of antitrust tying claims, but market power must be proven rather than be

⁷⁹³ Fellmeth, *supra* note 695, at 16.

⁷⁹⁴ *Mallinckrodt, Inc. v. Mediapark, Inc.*, 976 F.2d 700, 704 (Fed. Cir. 1992).

⁷⁹⁵ John Baker McClanahan, *Copyright Misuse As A Defense In An Infringement Action: Lasercomb America, Inc. v. Reynolds*, 49 WASH. & LEE L. REV. 213, 216, 231 (1992).

⁷⁹⁶ *Independent Ink, Inc. v. Trident, Inc.*, 210 F.Supp.2d 1155 (C.D. Cal. 2002), *aff'd in part, rev'd in part, sub nom. Independent Ink, Inc. v. Illinois Tool Works, Inc.*, 396 F.3d 1342 (Fed. Cir. 2005); *cert. granted*, 547 U.S. 28, 126 S. Ct. 1281 (2006) [hereinafter *Illinois Tool*] (Independent Ink sued Illinois Tool Works, claiming that one of its companies, Trident, had violated sections 1 and 2 of the Sherman Act by illegally tying its unpatented ink to its patented printheads. Trident makes ink and printheads, and requires anyone who uses Trident printheads to also purchase its ink. Independent Ink makes ink that is compatible with Trident's printheads).

presumed.⁷⁹⁷ Given the recent developments in case law the question remains whether the patent misuse doctrine retains any viability beyond its application to restrictions that continue to violate the antitrust laws. In other words, has the Patent Misuse Act finally overruled *Morton Salt*?

1.2.6.2. Copyright Misuse Doctrine

The essence of the copyright misuse defense is that the copyright owner should not be allowed to use the copyright to secure an exclusive right or limited monopoly not granted by, or otherwise inconsistent with the public policy embodied in copyright law.⁷⁹⁸ The argument for a defense to an infringement of copyright based on misuse⁷⁹⁹ is predominantly drawn upon the similarities between the interwoven, ever-evolving concepts and policies of the patent and copyright laws.⁸⁰⁰ Furthermore, the *Lasercomb* court concluded that the framers of the U.S. Constitution had considered the concepts of patent and copyright as similar enough in purpose to include these concepts in Article I, section 8, clause 8 of the Constitution of the United States. If the scope of copyright misuse may indeed be modeled after patent misuse, then statutory changes⁸⁰¹ and changing case law

⁷⁹⁷ *Illinois Tool*, 126 S. Ct. at 1291 (reasoning that “[w]hile some such arrangements are still unlawful, such as those that are the product of a true monopoly or a marketwide conspiracy, that conclusion must be supported by proof of power in the relevant market rather than by a mere presumption thereof”).

⁷⁹⁸ See, e.g., *Lasercomb*, 911 F.2d at 977-79.

⁷⁹⁹ In *Lasercomb*, a copyright case, the Fourth Circuit concluded that a misuse of copyright defense is inherent in the law of copyright just as a misuse of patent defense is inherent in patent law because of the similarity of rationales underlying the law of copyrights and the law of patents. In *Lasercomb*, the Fourth Circuit reiterated the Supreme Court repeatedly recognizing that both patent policy and copyright policy are designed to achieve advancement of public welfare by encouraging individual effort of authors and inventors in “Science and useful Arts”. See *Mazer v. Stein*, 347 U.S. 201, 219 (1954).

⁸⁰⁰ See *Morton Salt*, 314 U.S. at 494 (a patent infringement case laying in the dictum the foundation for analogizing copyright misuse to patent misuse).

⁸⁰¹ Some United States senators expressed doubt as to whether the copyright misuse doctrine existed and asserted that the passage of patent misuse legislation would not affect copyright law. 133 CONG. REC. S10275 (daily ed. July 21, 1987) (Senator Deconcini stated that “[t]itle XXIV deals only with patent law—not copyrights or trademarks or other intellectual property issues. Moreover, the Judiciary Committee has not had before it any proposal relating to copyright misuse in the 100th Congress, and therefore we had no pending proposal to address.” Senator Hatch agreed that “. . . through our approval of title XXXIV we are not sending any kind of positive message to courts who will be considering [the copyright misuse] doctrine” but assured later “that no one [shall] draw any negative implication from the fact that we have not restricted the application of the so-called copyright misuse doctrine in the same way we limit patent misuse. We do not want anyone to conclude that by not dealing with copyright misuse Congress has somehow, first, recognized the validity of this little-known doctrine, or second, given it an expanded application of meaning.”).

affecting the scope of patent misuse may be deemed to sway copyright misuse to the same extent.

Assuming the analogy of the patent misuse model to a copyright misuse context means that tying-in agreements threaten competition only if the copyright owner has sufficient economic power in the relevant market for the copyrighted work to force the sale or license of the tied products and thereby limit consumer choice in the market for the tied product. Here, the role of antitrust standard is said to provide the only available set of well-developed analytical tools to perform an appropriate analysis.⁸⁰²

Although *Illinois Tool* involved patents, the court's reasoning suggests that it shall reach the same conclusion with respect to copyrights and other intellectual property⁸⁰³ because the market power presumption for all forms of intellectual property derives from the patent misuse doctrine. Evaluating copyright misuse under the narrow antitrust standard adopted in *Illinois Tool* will result in limiting the doctrine's purview to cases in which the defendant is able to demonstrate plaintiff's market power. Contrary to a patent grant, the legal monopoly conferred by a copyright grant does not necessarily confer economic power. Consequently, we might question whether an analogous adoption of the antitrust standard to copyright misuse may be regarded appropriate.

Differences in terms of the actual grants of copyrights and patent rights nourish the discussion about the scope of copyright misuse.⁸⁰⁴ Whereas a patent grants a patent owner virtually complete control over the owner's invention, a copyright gives a

Against the backdrop of statutory development in patent law, no negative implication should be drawn from the fact that the patent misuse reform debate has dealt solely with patent law—not with copyrights in the same way: “We do not want anyone to conclude that by not dealing with copyright misuse Congress has somehow, first recognized the validity of this little-known doctrine, or second, given it an expanded application of meaning.” 133 CONG. REC. S10275 (daily ed. July 21, 1987) (statement of Senator Hatch).

⁸⁰² See generally Note, *supra* note 749, at 1302-3 (applying antitrust law in a copyright misuse context to secure free competition).

⁸⁰³ To support its broad holding the court cites agency enforcement guidelines and academic commentaries that expressly equal copyrights to other intellectual property in general. *Id.* at 1292-3 (referring to the 1995 Guidelines issued jointly by the Department of Justice and the Federal Trade Commission which state that in the exercise of their prosecutorial discretion the enforcement agencies “will not presume that a patent, copyright, or trade secret necessarily confers market power upon its owner”).

⁸⁰⁴ Frischmann & Moylan, *supra* note 684, at 876 (stating that “[t]he coordination of copyright and patent law is a more complicated task than courts want to make believe us because “it involves important policy decisions as to the appropriate social cost-benefit trade-off for promoting development of different types of innovation” (citations omitted)).

copyright owner only limited control in that not the idea itself but only an author's expression of an idea is protected through sharply delineated exclusive rights.⁸⁰⁵ Thus, many expressions of the same or similar idea may be available on the market, making the copyrighted material more substitutable and less likely to produce an anti-competitive effect.⁸⁰⁶ Thus, in determining whether a copyright constitutes misuse "[t]he question is not whether the copyright is being used in a manner violative of antitrust law...but whether the copyright is being used in a manner violative of the public policy embodied in the grant of copyright."⁸⁰⁷

The patent model has generally been concerned with monopolistic practices in ways that violate the antitrust laws because a patent confers utterly comprehensive control over the patented item. Therefore, a patent may be considered misused if it is used in a way that violates the antitrust laws or when it is used to expand the scope of the patent rights in an anti-competitive fashion.⁸⁰⁸ Given the different nature of patents and copyrights, anti-competitive conduct thus cannot justify invoking an affirmative misuse defense as easily to copyright law as it does to patent law. However, with view to the increasing trend towards standardization of computer software appearing hand in hand with the rise of software monopolization in the computer industry, like in the case of Microsoft products, a pure antitrust standard would seem an appropriate tool for the end of achieving interoperable products and so spurring competition.

The program's copyright holder can either win or lose the battle under the fair use analysis. An extension of the fair use doctrine into the area of decompilation would leave software providers powerless to stop the use of their copyrighted computer software by others, especially competitors, to plunder profits that would otherwise belong to the software providers. Otherwise a broad understanding of fair use as to apply to software decompilation would leave software providers with copyright-like protection of the functional elements of their software simply because the

⁸⁰⁵ 17 U.S.C. § 106(1)-(5) (enumerating exclusive rights granted to copyright owners); 17 U.S.C. § 106A (rights granted to authors)—both subject to limitations set forth in 17 U.S.C. §§ 107 through 122.

⁸⁰⁶ McClanahan, *supra* note 795, at 231.

⁸⁰⁷ See, e.g., *Lasercomb*, 911 F.2d at 978; *Practice Management*, 121 F.3d at 521.

⁸⁰⁸ See *Windsurfing Int'l, Inc. v. AMF, Inc.*, 782 F.2d 995 (Fed.Cir. 1986).

utilitarian aspects of the software are not directly observable.⁸⁰⁹ By the alternative legal vehicle of the copyright misuse defense⁸¹⁰ overruling the fair use case law which is generally in favor of reverse engineers the United States are moving a step closer to the realization of the U.S. model of a “closed system” as it has recently appeared to be adopted.

1.3. Contract Law and Decompilation

From a contract law perspective, it must also be recognized that the decompilation right under Article 6 of the EU Software Directive is an exception which cannot be altered by contract.⁸¹¹ Again, the situation is different under U.S. copyright and contract law. Licensing restrictions prohibiting decompilation and reverse engineering are not *per se* illegal but their enforceability is presumably governed by the rule-of-reason. If the copyright holder cannot establish a legitimate reason for the prohibition against reverse engineering, the infringement suit might be threatened to turn into an anti-competitive attempt to prohibit the development of new software products.⁸¹²

In the United States, the reasoning and conclusions of the *Vault* ruling have not been adopted by other circuits.⁸¹³ Furthermore, the U.S. Supreme Court denied to *certiorari* petitions for Bowers,⁸¹⁴ and the Court was not asked to grant *certiorari* for Davidson.

⁸⁰⁹ See Williams, *supra* note 445, at 257 (identifying the fair use exception as the only viable means for accessing a software’s uncopyrightable, but hidden, ideas and functions).

⁸¹⁰ See *id.* (pointing out drawbacks of the fair use doctrine in the software reverse engineering context but without considering the copyright misuse defense as alternative legal basis).

⁸¹¹ CZARNOTA & HART, *supra* note 52, at 126.

⁸¹² Lewis, *supra* note 91, at 586.

⁸¹³ 1 NIMMER ON COPYRIGHT, *supra* note 15, § 1.01[B][1][a][iii], at 1-27. The U.S. approach is not at all uniform by and in itself. See *Bowers*, 320 F.3d at 1323 (holding the shrink-wrap license prohibiting reverse engineering enforceable); *ProCD*, 86 F.3d at 1449 (holding shrink-wrap licenses “enforceable unless their terms are objectionable on grounds applicable to contracts in general”); *but see Vault*, 847 F.2d at 269 (holding license term unenforceable because Louisiana law allowing licenses prohibiting adaptation using decompilation or disassembly conflicts with federal copyright law).

⁸¹⁴ 539 U.S. 928 (2003). According to the petition for certiorari, shrink-wrap licenses typically provide that, just by opening a box of software or by using the software, the user agrees to the license terms. The petition stated that reverse engineering has long been regarded as fair use, and thus not an infringement of copyrights, because it promotes innovation and progress. As a consequence of the Federal Circuit’s decision, prohibiting anyone who lawfully possesses a

Another strategy was adopted by U.S. law making authorities. The Uniform Computer Information Transactions Act (UCITA),⁸¹⁵ a proposed uniform state law that seeks to settle the law regarding contracts and licenses in the information technology realm.⁸¹⁶ The UCITA contains a specific provision stating that license terms in contradiction to federal law are unenforceable.⁸¹⁷ The European Communities also expressed doubt over the legal status of shrink-wrap licenses:

Nor is it entirely clear whether the practice of so-called ‘shrink-wrap licensing’ where use conditions are attached to a product which is, to all intents and purposes ‘sold’ to the user, constitutes a valid licen[s]e in all circumstances and in all jurisdictions.⁸¹⁸

The Commission well recognized the need to distinguish between different modes of commercial exploitation: by either an outright sale or licensing. While by “sale” certain mandatory rights to use the program pass to the purchaser along with the physical copy of the computer program,⁸¹⁹ by way of “licensing” the rights to use the program, with a limited number of exceptions, are amenable by deviant contractual arrangements.⁸²⁰ However, the European legislators did not discuss any further legal consequences of the recognized distinction.

competitor’s computer software from discovering unprotected elements and functionality through reverse engineering will chill innovation and progress in the software industry.

⁸¹⁵ See Uniform Computer Information Transactions Act (revised and amended in 2002) [hereinafter UCITA], available at <http://www.law.upenn.edu/bll/ulc/ucita/2002final.htm> (last visited April 25, 2007).

The UCITA began as a proposal drafted by the National Conference of Commissioners on Uniform State Law (NCCUL) to change article 2B of the Uniform Commercial Code. After abandonment of this project, the UCITA was proposed as a stand-alone uniform act in July 1999.

⁸¹⁶ See Lydia Pallas Loren, *Slaying the Leather-Winged Demons in the Night: Reforming Copyright Owner Contracting with Clickwrap Misuse*, 30 OHIO N.U. L. REV. 495, 496 n.4 (2004) (stating the minor significance of the UCITA while the UCITA was adopted in only two States, Maryland and Virginia and ultimately, in February 2003, the UCITA was withdrawn from consideration by the ABA House of Delegates).

⁸¹⁷ UCITA § 105(a) (West 2006) reads in full:

“[Federal preemption.] A provision of this [Act] which is preempted by federal law is unenforceable to the extent of the preemption.”

The UCITA does not expressly authorize anti-reverse engineering provisions in license agreements. However, it arguably does so implicitly with its broad authorization of mass-market license agreements which routinely contain such terms. See UCITA § 209 (West 2006).

⁸¹⁸ *Commission Preparatory Memorandum*, supra note 24, para. 3.4., at 7.

⁸¹⁹ The Commission apparently indicates the “Community exhaustion doctrine”, or the doctrine of “first sale” as called in the U.S. jurisdiction. The first sale of a copy of a computer program in the Community market by the copyright holder or with his consent exhausts the exclusive rights conferred by the original program within the Community as regards the copy thereof. EU Software Directive, supra note 10, art. 4(c), at 44.

⁸²⁰ See *Commission Preparatory Memorandum*, supra note 24, para. 3.5., at 7.

In conclusion, the principally more liberal, but inconsistent, U.S. approach as adopted in recent case law leaves open the possibility that anti-reverse engineering clauses may be enforceable under state contract law. Hence, U.S. software owners may likely be tempted to press for stronger shrink-wrap and/or click-on contracts.⁸²¹ In such circumstances, the European software industry finds itself at a disadvantage compared to the competitive U.S. software market. While all software products, domestic as well as foreign, will be equally vulnerable to decompilation and reverse engineering in Europe, U.S. software providers will also continue to enjoy a vast U.S. market that affords their products greater protection. Put into other words, U.S. software producer may get stimulated by the ideas embodied in European computer programs and which may be lawfully decompiled there, while across the Atlantic computer programs are kept in the U.S. software providers' vaults which are locked vis-à-vis European would-be reverse engineers by means such as legally restrictive license agreements.

At the same time European providers will face the greatest threat of legally sanctioned appropriation of their confidential programming techniques right in their own home market. All in all, the incentive for the European software industry to invest in innovation would be reduced, resulting in a less competitive position for European software providers in world markets.⁸²² With the legal outcome of the EU Software Directive to be enacted in response to the U.S. software industry's dominance at the European market⁸²³ the European legislators appear to have failed their objective in that regard. Also under practicability considerations, it may be costly and impractical for software providers generally to offer and enforce one set of contractual terms in the United States and a different set in the European Communities. For all of these reasons, global uniformity of the governing law would enhance Europe's position in the world software markets and particularly pose a counterpart to the U.S. leading position at these markets.

⁸²¹ See Jennifer R. Knight, *Copyright Misuse v. Freedom of Contract: And the Winner Is...*, 73 TENN. L. REV. 237, 266 (2006) (predicting an increase of "private legislation").

⁸²² Haaf, *supra* note 23, at 419.

⁸²³ See O'Rourke, *supra* note 601, at 552-3.

With the now author-friendly U.S. copyright law in mind, European software copyright holders when acting in trans-atlantic dimensions may want to set up choice-of-U.S. law-clauses to ensure enforceability of their anti-decompilation clauses and in so doing to escape from Article 9(1) of the EU Software Directive. Would this mean that the European legislators and end-users have lost the decompilation battle? At least for the time being, they are stuck with the hypothesis that market is expected to intervene in a regulatory manner if anti-reverse engineering clauses in shrink-wrap and click-on license agreements are truly detrimental to technological innovation and Europe's competitiveness likewise. The market is thought to come up with an alternative to restrictive licensing practices inhibiting decompilation and reverse engineering. The hypothesis, for example, has already proved to work true when open source software providers emerged on the market.⁸²⁴ Nonetheless, the question remains: Will the reverse engineering community be satisfied with the "comfort" of a free market approach?

2. *The Future: Compulsory Licensing of Interface Information as Sui Generis Copyright Remedy?*

2.1. Unconventional Consequences of Unconventional Circumstances

Copyright only protects expression, and only protects expression from reproduction and other acts of exploitation. The initial software programmer owns some exclusive rights over his program's creative expressions, while leaving others free to study the expression and even to copy the ideas and principles embedded in the copyrighted computer program. In terms of achieving interoperability, it is not the program code itself (i.e. its expression) which is required, it is the way the interface works (i.e. its ideas and principles) which is essential.⁸²⁵ Uncopyrightability of ideas and principles is a principle which is well-accepted in copyright law. Notwithstanding this general copyright rule the question whether copyright gives also a right of access to interface information embodied in a copyrighted computer

⁸²⁴ Cf. Pruitt, *supra* note 533, at 66.

⁸²⁵ BAINBRIDGE, *supra* note 4, at 117-8.

program is far from a unanimous response.⁸²⁶ Current legal regimes traditionally draw a distinction whereby copyright law is to define the scope of a software developer's control over his software, while it falls within the purview of competition law to determine when and how access to ideas and principles underlying a copyrighted work, such as interface specifications of a computer program, should be granted.⁸²⁷ This Part re-considers this conventional distinction in light of the hybrid structure of computer programs.

Two strong arguments doubt the traditional distinction to be suitable where computer programs are concerned. First, copyright law essentially protects expression, not function. With computer software, however, function is certainly more important than form⁸²⁸ which would lead to the conclusion to protect also access to software functions, such as interface information. Second, without the ability to study copyrighted software in search for how its functional elements (i.e. the interface specifications of a computer program) work, other software programmers would be unable to develop compatible products.⁸²⁹ Due to the unique design of computer programs, access may require methods of decompilation, which necessitates, as an intermediate step, copying object code into a computer's memory. Interim copying of the target computer program is *conditio sine qua non* for access to its underlying (functional) interface information.

Finding that the use of copyrighted computer elements in the process of reverse engineering is unlawful because it intrinsically infringes the software author's exclusive rights in his software would give the software copyright owner a *de facto*

⁸²⁶ Thomas F. Cotter, *Intellectual Property and the Essential Facilities Doctrine*, in RUTH TOWSE & RUDI HOLZHAUER (EDS), *THE ECONOMICS OF INTELLECTUAL PROPERTY—VOLUME IV* (2002), at 185 (concluding that under both the Lockean and the utilitarian theory the exclusivity of some copyrights may not be understood as to render those rights absolute in preventing others from having access to those creations under all circumstances but rather a series of limitations should create sufficient access and incentive to continue to create new works).

⁸²⁷ See, e.g., Rotenberg, *supra* note 128, at 6; MORITZ & TYBUSSECK, *supra* note 16, at 71.

⁸²⁸ Cf. *Sega*, 977 F.2d at 1524 (discussing the nature of computer programs as to be basically utilitarian in function because the programs are used to operate machines to get results).

⁸²⁹ Intellectual property rights are the engine and guaranty of free and undistorted competition in the form of innovative competition. The U.S. Constitution pursued this goal in Article I, section 8, clause 8 as “[t]o promote the Progress of Science and useful Arts by securing for limited Times to Authors and Inventors the exclusive Rights to their respective Writings and Discoveries.” The rationale—not only underlying the U.S. constitutional Intellectual Property Clause but generally modern intellectual property regimes worldwide—is to serve the public welfare. MORITZ & TYBUSSECK, *supra* note 16, at 151.

monopoly over the functional elements of its program for as long as its copyright lasted.⁸³⁰ In order to obtain a lawful monopoly on the functional concepts of a computer program, the software developer may seek a patent grant provided that the software satisfies the more stringent standards of the patent laws.⁸³¹ In other words, to allow protection of interface information under copyright law by way of refusing access to that interface information would amount to introducing an element of unfair competition into copyright law that would render copyright law comparable with patent law, however, without the same safeguard measures for the public interest.⁸³²

Software interoperability walks the thin line between copyright and antitrust law. As a consequence of these “unconventional circumstances” as it is the case with computer software, in 2007, Europe has once again led the global software industry by relying on “unconventional consequences” in the *Microsoft* case on how to deal with computer software at the crossroads between copyright law and antitrust law. The Court of First Instance (CFI) introduced a compulsory licensing scheme for computer source code. At the other side of the Atlantic, recent U.S. case law has carved out the doctrine of copyright misuse in analogy to the patent misuse and in so doing reached the same crossroads.⁸³³ Unlike the CFI, U.S. case law has not yet come up with a distinct and uniform response of how to position the doctrine of copyright misuse in the reverse engineering context.

2.2. Interoperability at the Crossroads of Copyright and Antitrust Laws

Computer software is an industrial tool which is essential to the Community’s economic developers to invest their intellectual and financial resources and thereby to contribute to technical progress in the public interest. Technical progress and

⁸³⁰ Therefore, at the very beginning of the legislative drafting of the EU Software Directive, the Commission assigned the problem of decompiling computer software to be regulated by competition law. See LEHMANN, *supra* note 2, at 18-9 (citing the Commission Preparatory Memorandum); see also MORITZ & TYBUSSECK, *supra* note 16, at 71 n.192 (attempting to clarify the Commission’s position insofar as the Commission conclusion on the adoption of the Proposed Directive should have been read as follows: “Access to [unpublished] information is not a matter of copyright law”).

⁸³¹ See, e.g., *Sony*, 203 F.3d at 605.

⁸³² Spoor, *supra* note 39, at 1081.

⁸³³ For a discussion see Part VII.1.2.4.

public welfare are also ensured by a system of undistorted competition being one of the principal goals of the EC Treaty.⁸³⁴ The way of how to accomplish these goals is called interoperability of computer software.

The software industry is dominated by two different legal regimes but one and the same problem. In the United States, the use of copyrights to protect the functional elements embodied in software, particularly interface information, has allowed rightholders to exert their rights in ways that could not have been imagined by the drafters of the original Copyright Act in 1976. Many of these potentially undesirable and illicit exertions have been banned either by fair use or less frequently under the doctrine of copyright misuse.⁸³⁵ In Europe, the protection of interface information became a subject of intense lobbying of computer software industry cycles during the preparatory stages of the EU Software Directive. The main thrust was the degree to which copyright protection was to cover interfaces because it was this very aspect which was to define the leading industry players' power to dominate the hardware and software market in the years to come. An excessive protection of interfaces could extinguish further progress towards interoperability and lead to abuses of the leading manufacturer's dominant position. Thus, the question of the extent of disclosure of program interfaces reaches not only for copyright considerations but also for competition aspects since some kind of access to interface information is indispensable to deprive a dominant supplier of refusing from making information available which is necessary for interoperability.⁸³⁶ In other words, a dominant supplier who refuses to make its interface specifications available to provide for interoperability between programs (or between programs and hardware) can also be subject to the application of the competition rules under Articles 81 and 82 of the EC Treaty.⁸³⁷ If the Directive had not provided for at least some kind of free access to interface information, the copyrightholders otherwise would have been able to pursue bundling strategies

⁸³⁴ See *Proposed Directive*, *supra* note 31, at 16.

⁸³⁵ The recognition of the copyright misuse defense has not yet been fully affirmed by the U.S. judiciary.

⁸³⁶ Cotter, *supra* note 826, at 194 (noting that most commercially successful intellectual property rights probably confer some "supracompetitive" profit, while among them a few may even give rise to monopoly power); see also EU Software Directive, *supra* note 10, recital 27, at 43.

⁸³⁷ The Commission did not discuss antitrust aspects of the EU Software Directive for nine years following its adoption.

which would hinder competition in the data processing industry, a fact which the Commission has always tried to prevent.⁸³⁸

According to the European understanding, the development of compatible programs is regarded to be desirable also from the point of view of industrial policy. In this regard, the Commission pointed out the negative consequences for communications and industry at large that would be triggered off if technically available access protocols and interface information were available only at a license fee that only the largest of competitors could afford, or if competitors were prevented from integrating into their product range protocols and interface information that are gaining wide support as international standards.⁸³⁹ Copyright could create a monopoly not only of the access protocol itself but of the entire segment of the systems market that depends on its use.⁸⁴⁰ In any event, situations which might cause an abuse of a dominant position may be established under the antitrust rules⁸⁴¹ as laid down in Articles 81 (ex-Article 85) and 82 (ex-Article 86) of the EC Treaty.

In the conclusions following the text of the Proposed Directive the Commission paid particular attention to the relation between the Community's antitrust rules and copyright. In principle, exclusive proprietary copyrights and free competition are designed to achieve the same objective by different means:

Any arrangement or measure which goes beyond the existence of copyright can be subject to control under the competition rules. This means that for example any attempt to extend by contractual agreements or other arrangements the scope of protection to aspects of the programs for which protection under copyright is not available, or the prohibition of any act which is not reserved for the right owner may constitute an infringement of the competition rules. Moreover, companies in a dominant position must not abuse that position within the meaning of Article [82] of the [EC] Treaty. ... Furthermore, the ability of a competing manufacturer to write an independent but compatible program often depends on his possibility to have access to the target program or to certain information relating to it. Access to information is not a matter of copyright law. Article [82] always applies where a dominant

⁸³⁸ Dreier, *supra* note 54, at 323-5.

⁸³⁹ In particular, small and medium software manufacturers joined in the ECIS raised these arguments in the decompilation debate in the course of the legislative drafting process.

⁸⁴⁰ See *Commission Green Paper*, *supra* note 47, at 193.

⁸⁴¹ See EU Software Directive, *supra* note 10, recital 27, at 43; *Commission Green Paper*, *supra* note 47, at 193.

company abusively refuses access to such information or restricts unreasonably such access.⁸⁴²

The Commission indicates the problem of applying the essential facility doctrine to copyrights given the competing policy objectives of copyright law and competition law. In this context the Commission refers to the distinction between the existence and the exercise of intellectual property rights as governed thus far by the European Court of Justice (ECJ). The exercise of copyright is not exempted from the scope of European competition law.⁸⁴³ Any arrangement or measure which goes beyond the intended purpose of the statutory exclusive rights is prone of being subject to control under the competition rules. In other words, if decompiling the target interface code ultimately fails competition considerations may become an issue. In software industry the question practically will be at which side of the Atlantic reverse engineers will be more likely to fail in their efforts to decompile computer software on legal grounds.

In terms of copyright law, the European approach as laid down in Article 6 of the EU Software Directive is a compromise that may disfavor, to some extent, the proprietors of copyrights in standard software, such as the Windows operating system program of Microsoft. Under U.S. copyright law, the fair use approach and the increasingly liberal standard of freedom of contract has recently begun to favor the owner of copyrighted software, thereby showing a tendency towards a “closed software” model. Under competition considerations, competition concerns may arise in either jurisdiction if copyright law cannot prevent a software owner holding the facility (i.e., interface specification in the form of object code) from refusing to provide other software developers with access to something that is vitally important to competitive viability in a particular market. Thus, extreme abuses of dominant position can, in any event, be limited through recourse to EU and national or U.S. antitrust rules respectively.⁸⁴⁴

⁸⁴² See *Proposed Directive*, *supra* note 31, at 16.

⁸⁴³ Roman Heidinger, *The Essential Facility Doctrine and Intellectual Property: Compulsory Licenses under EC Competition Law*, MR-Int 2006, 23.

⁸⁴⁴ VAN DEN BERG & CAMESASCA, *European Competition Law and Economics: A Comparative Perspective* (2001) at 272.

The Commission noticed the respective tension between copyright law and competition law⁸⁴⁵ and henceforth stated that the exercise of exclusive copyrights will not prejudice the application of the competition rules and the imposition of effective remedies in appropriate cases.⁸⁴⁶ However, a conflict may occur where a copyright owner is in a position to exercise his statutory exclusive rights beyond their intended purpose. Exclusive copyrights may procure a dominant position of the software provider holding the copyright in the software. The existence of such dominant position is lawful so long as the position does not amount in an abuse within the meaning of Article 82 (ex-Article 86) EC Treaty. As example, the Commission quotes that under certain circumstances the exercise of copyright as to the aspect of information contained in a program, which other companies need to use in order to write compatible programs, could constitute such an abuse.⁸⁴⁷ The same outcome will be reached in the case that a dominant company tries to use its exclusive rights in one product to gain an unfair advantage in relation to one or more products not covered yet by these rights (“tying-in”). Exactly this point of concern became in the course of the legislative drafting, and actually still is, the target of the worldwide decompilation debate. Here, the distinctive feature is the impact of the refusal to deal on competition, the so-called “essential facility doctrine”.

So far we were concerned with the copyright aspects under EU and U.S. law of making source code available through the process of decompilation. The ability of a competing manufacturer to write an independent but compatible program often depends on his possibility to have access to the target program or to certain information relating to it. Access to information is not a matter of copyright law but Article 82 (ex-Article) of the EC Treaty applies where a dominant company abusively refuses access to such information or restricts unreasonably such access.

⁸⁴⁵ LEHMANN, *supra* note 2, at 2 (indicating that the Commission tried to reach a counterbalance between adequate monopoly-like copyright protection on the one hand and a system of free competition on the other hand).

⁸⁴⁶ See *Proposed Directive*, *supra* note 31, at 16.

⁸⁴⁷ In *Volvo v. Veng*, Case C-238/87, [1988] ECR 6039, the ECJ held that the mere existence of protected design could not in itself constitute an abuse of a dominant position. However, the exercise of such an exclusive right by the owner of the registered design may be prohibited by Article 82 of the EC Treaty if it involves certain abusive conduct, such as the arbitrary refusal to supply spare parts to independent repairers, the fixing of prices for spare parts at an unfair level or a decision no longer to produce spare parts for a particular model that is still in circulation.

At the end of the day we can draw the conclusion that copyright law does not go far enough in ensuring adequate access to copyrighted program code. In the following, the paper will discuss whether competition law—in the form of the essential facility doctrine—provides a vivid alternative to decompilation for competing software developers whom access to another program’s source code is denied and therewith access to the relevant market at whole. In other words, perhaps the essential facility doctrine can be viewed as a method for imposing a duty to share one’s copyrighted interface specification for the purpose of achieving interoperability, in cases in which copyright law does not, but should, impose such an obligation, that is a compulsory licensing order.

European authorities attempt to bridge the tension between copyright law and competition law⁸⁴⁸ emerging with regard to essential facilities by applying the essential facility doctrine to copyright. In this regard, the recent decision of the CFI in the antitrust case of *Microsoft* and its significant, possibly transatlantic, implications on the legal status of software authorship and consequently on software development will be discussed in detail for the purpose of this paper.

2.3. The General Concept of the Essential Facility Doctrine

Basically, it is deemed pro-competitive to allow companies the exclusive use of which they have acquired or construed.⁸⁴⁹ In principle, this applies also to undertakings with considerable market power. The essential facility doctrine is an exemption to the general rule of the freedom of (dominant) companies to choose their business partners.⁸⁵⁰ Where a dominant company owns an “essential” or “bottleneck” asset that company may be required by law to share or deal (i.e.

⁸⁴⁸ *But see* MORITZ & TYBUSSECK, *supra* note 16, at 151 (stating that despite the fact that there is a tension between the subject matter of both legal regimes their relationship to each other is sometimes characterized as complementary rather than antagonistic because both copyright law and competition law pursue the policy of maximizing consumer welfare, however, each cuts its own way to achieve that goal).

⁸⁴⁹ Heidinger, *supra* note 843, at 23.

⁸⁵⁰ See Wolfgang Schuhmacher, *Die Essential Facility Doctrine als Herausforderung für das Urheberrecht [The Essential Facility Doctrine as Challenge for Copyright Law]*, at 101-2 (2003), distinguishing general “Kontrahierungszwang” from the essential facility doctrine as follows: “Kontrahierungszwang” serves to protect the weaker against the dominant market players, while the purpose of the essential facility doctrine is to open the market to potential entrants.

provide access to) that asset at a reasonable price and on fair terms.⁸⁵¹ The doctrine is justified as to be of pro-competitive effects because if it comes to the remedy (dominant) companies will be obliged to give access to competitors if the refusal to do so has sufficiently serious effects on competition.⁸⁵² The original concept of an essential facility is originated in U.S. antitrust case law.⁸⁵³

Recital 27 of the EU Software Directive states that “the provisions of the Directive are without prejudice to the application of the competition rules under Articles [81 and 82 of the EC Treaty] if a dominant supplier refuses to make information available which is necessary for interoperability as defined in this Directive.”⁸⁵⁴ Whereas the *existence* of copyright protection is ruled by the EU Software Directive itself, *access* to information necessary to accomplish interoperability is ruled, essentially, by EC competition law.⁸⁵⁵

2.3.1. U.S. Antitrust Law

The core legislative provisions of U.S. antitrust law are laid down in Sections 1 and 2 of the Sherman Act.⁸⁵⁶ Section 1 proscribes “[e]very contract, combination in the form of trust or otherwise, or conspiracy, in restraint of trade or commerce among the several States, or with foreign nations.” Case law pertaining to Section 1 is often concerned with finding an improperly restrictive agreement as one person cannot contract, combine, or conspire alone. On the other hand, Section 2 proscribes every person from monopolizing, or attempting to monopolize, or combining or conspiring with any other person(s) to monopolize any part of the trade or commerce among the several States, or with foreign nations. Section 2 principally applies to unilateral conduct that is the creation or misuse of monopoly power through wrongfully exclusionary means. Section 2 does not forbid

⁸⁵¹ See SIMON BISHOP & MIKE WALKER, *The Economics of EC Competition Law: Concepts, Application and Measurement*, para. 6.104 et seq., at 238 et seq.; see also Cotter, *supra* note 826, at 196-7.

⁸⁵² Heidinger, *supra* note 843, at 23.

⁸⁵³ For a historic outline concerning the development of the essential facility doctrine in the United States see, e.g., Cotter, *supra* note 826, at 197-8.

⁸⁵⁴ EU Software Directive, *supra* note 10, recital 27, at 43.

⁸⁵⁵ MARLY, *supra* note 111, at 304.

⁸⁵⁶ Sherman Antitrust Act, 15 U.S.C. §§ 1-7 (1890).

monopoly *per se*, but bars monopolization as well as attempts and conspiracies to monopolize. Thus, section 2 case law focuses not on the fact of monopoly alone, but rather on how a monopoly has been gained or sustained.

With intellectual property federal enforcement agencies and the courts have now begun to recognize that the intellectual property and antitrust laws are not mutually inconsistent but that both are directed at the enhancement of consumer welfare.⁸⁵⁷

That is also reflected in the “Antitrust Guidelines for the Licensing of Intellectual Property”,⁸⁵⁸ which were issued in 1995 in response to the increasing importance of intellectual property in the growth of the U.S. and world economics and the need for guidance as to how the antitrust laws will be applied to intellectual property licensing.⁸⁵⁹ The IP Guidelines do not presume that possession of intellectual property creates market power⁸⁶⁰ given their generally pro-competitive nature. In a rule-of-reason analysis the licensing arrangements and their potential anti-competitive effects are to be evaluated.⁸⁶¹ In the interests of providing some measure of certainty, intellectual property licensing agreements will not challenge antitrust law at all (“antitrust safety zone”), if (i) the restraint is not facially anti-competitive,⁸⁶² and (ii) the licensor and its licensees collectively account for no more than 20 per cent of each relevant market significantly affected by the restraint.⁸⁶³ The rules set forth in the IP Guidelines do not make any difference in being applied to different kind of property.⁸⁶⁴

A third way, besides sections 1 and 2 of the Sherman Act, to encounter a refusal to deal or to share an asset with another market player is the application of the

⁸⁵⁷ ERNEST GELLHORN, ET ALIA, ANTITRUST LAW AND ECONOMICS IN A NUTSHELL (2004), at 477.

⁸⁵⁸ See IP Guidelines, *supra* note 769.

⁸⁵⁹ See generally DOUGLAS F. BRODER, A GUIDE TO US ANTITRUST LAW (2005) 112-3.

⁸⁶⁰ See IP Guidelines, *supra* note 769, § 2 and § 5.3. (“The Agencies will not presume that a patent, copyright, or trade secret necessarily confers market power upon its owner”); see also *In re Independent Service Organizations Antitrust Litigation*, 203 F.3d 1322 (Fed. Cir. 2000) (“A patent alone does not demonstrate market power.”).

⁸⁶¹ See *id.*, § 4 (pointing out that even if anti-competitive effects are likely, they may be counterbalanced by pro-competitive effects such as efficiencies or by mitigating factors, such as a relatively short duration).

⁸⁶² See *id.*, § 4.3. n.31 (defining “facially anticompetitive” restraints as those that would normally be viewed *per se* illegal).

⁸⁶³ See *id.*, § 4.3.

⁸⁶⁴ Schuhmacher, *supra* note 850, at 92-3.

essential facility doctrine.⁸⁶⁵ In relying on Section 2 a person who owns a facility that cannot practically be duplicated by would-be competitors is required to share it on fair terms.⁸⁶⁶

2.3.2. EC Competition Law

Article 82 of the EC Treaty is the principal substantive rules dealing with anti-competitive behavior regarding the refusal to supply and the grant of access to essential facilities.⁸⁶⁷ The ECJ has repeatedly emphasized that the notion of abuse is a flexible notion and that the list of examples in Article 82 is non-exhaustive.⁸⁶⁸ The most contentious issue in the application of Article 82 to licensing involves the circumstances where a dominant licensor may be obliged to grant a license. If there is no potential anti-competitive link to the refusal to supply, then access will unlikely be mandated.^{869 870}

⁸⁶⁵ The first U.S. case dealing with the essential facility doctrine was *United States v. Terminal Railroad Ass'n St. Louis*, 224 U.S. 383, 392 (1912).

⁸⁶⁶ Under U.S. case law applying the essential facility doctrine, an asset that either constitutes or controls access to a specific market may qualify as an "essential facility" if the following elements are proven: (i) control of the essential facility by a monopolist or one who has close to a 100 per cent market share and the barriers to entry are very high; (ii) a competitor's inability practically or reasonably to duplicate the essential facility; (iii) the denial of the use of the facility to a competitor; and (iv) the feasibility of providing the facility. *MCI Communications Corp. v. American Tel. & Tel. Co.*, 708 F.2d 1081, 1132-3 (7th Cir. 1983).

⁸⁶⁷ Article 82 of the EC Treaty, at 65, reads in full text:

"Any abuse by one or more undertakings of a dominant position within the common market or in a substantial part of it shall be prohibited as incompatible with the common market in so far as it may affect trade between Member States. Such abuse may, in particular, consist in:

(a) directly or indirectly imposing unfair purchase or selling prices or other unfair trading conditions;

(b) limiting production, markets or technical development to the prejudice of consumers;

(c) applying dissimilar conditions to equivalent transactions with other trading parties, thereby placing them at a competitive disadvantage;

(d) making the conclusion of contracts subject to acceptance by the other parties of supplementary obligations which, by their nature or according to commercial usage, have no connection with the subject of such contracts."

⁸⁶⁸ See, e.g., Case C-333/94 P *Tetra Pak International SA v. Commission* [1996] ECR I-5951, [1997] 4 CMLR 662, at paragraph 37 of the judgment, where the ECJ held: "It must, moreover, be stressed that the list of abusive practices set out in the second paragraph of Article 86 [now Article 82] of the Treaty is not exhaustive. Consequently, even where tied sales of two products are in accordance with commercial usage or there is a natural link between the two products in question, such sales may still constitute abuse within the meaning of Article 86 [now Article 82] unless they are objectively justified."

⁸⁶⁹ BISHOP & WALKER, *supra* note 851, para. 6.113, at 244 (listing examples, e.g. concerns over the credit-worthiness of the entrant or a lack of spare capacity, for entirely legitimate reasons for refusing access).

2.4. The Essential Facility Doctrine Applied to the Incentive/Access Tradeoff

The ownership of an intellectual property right is not in itself an abuse of a dominant position rather the use of such a right may amount to one.⁸⁷¹ The holder of an intellectual property right can always be considered to be a monopolist, whose monopoly is, moreover, protected by intellectual property law.⁸⁷² With copyright, the grant of copyright in a software provider's program code (i.e., interface specification) encourages innovation by rewarding the copyright owner with exclusivity, but that may in turn become statutory dominance which may be abused.⁸⁷³ The monopoly inherent in the exclusive copyrights equals a refusal to supply which may result in enabling the copyright owner to hold a dominant position in the market. If that dominance extends to a substantial part of the internal market, the copyright holder will enjoy a position of dominance.⁸⁷⁴

It is the ECJ's policy that a refusal to license is not abusive *per se*, however the concurrence of certain circumstances might make it so.⁸⁷⁵ Here, the essential facility doctrine as a subcategory of Article 82⁸⁷⁶ comes into play. EC competition law has found a refusal to supply to contravene Article 82 of the EC Treaty. In one of the early cases in this field, in *Commercial Solvents*,⁸⁷⁷ the ECJ held that

⁸⁷⁰ STEVEN D. ANDERMAN & JOHN KALLAUGHER, *TECHNOLOGY TRANSFER AND THE NEW EU COMPETITION RULES* (2006), para. 10.33, at 280.

⁸⁷¹ Competition law has always been recognizing exclusive intellectual property rights as essential ingredient to spice up competition at the market. *See, e.g.*, Case C-24/67 *Parke, Davis & Co. v. Probel, Reese, Beintema-Interpharm and Centrafarm* [1968] ECR 55, [1968] CMLR 47.

⁸⁷² ALBERTINA ALBORS-LLORENS, *EC COMPETITION LAW AND POLICY* (2002) 100-1.

⁸⁷³ BARRY J. RODGER & ANGUS MCCULLOCH, *COMPETITION LAW AND POLICY IN THE EUROPEAN COMMUNITY AND UNITED KINGDOM* (1999) 101.

⁸⁷⁴ As the ECJ indicated in *Deutsche Grammophon Gesellschaft m.b.H. v. Metro-SB-Großmärkte*, Case C-78/70 [1971] ECR 487, [1971] 1 CMLR 631, only if this dominance extended to a "substantial part of the Common Market", the undertaking would hold a dominant position within the meaning of Article 82.

⁸⁷⁵ *See, e.g.*, Case C-238/87 *AB Volvo v. Erik Veng (UK) Ltd.* [1988] ECR 6211, [1989] 4 CMLR 122 [hereinafter *Volvo*], at paragraph 9 of the judgment, where the Court stated that certain circumstances, such as the arbitrary refusal to supply spare parts or the charging of unreasonably high prices for these spare parts, could render that refusal abusive.

⁸⁷⁶ *Schuhmacher*, *supra* note 850, at 89.

⁸⁷⁷ *See Institutio Chemioterapico Italiano Spa and Commercial Solvents Corp. v. Commission – Joined Cases 6 & 7/73* [1974] ECR 223, [1974] 1 CMLR. 309 [hereinafter *Commercial Solvents*].

In *Commercial Solvents*, an American company made two products used for the manufacture of a drug to cure tuberculosis. Its Italian subsidiary refused to supply one of these products to a long-standing customer that manufactured the anti-tuberculosis drug. This refusal to supply took place shortly after *Commercial Solvents'* Italian subsidiary itself started manufacturing the same drug in

refusing to supply a downstream competitor⁸⁷⁸ in order to restrict competition in the market for the final product must be considered an abuse within the meaning of Article 82 and ordered the company to resume supplies. The basic rationale behind the essential facility doctrine is that one competitor in a downstream market must not be able to get control over the only source of supply of an input which is essential in that market,⁸⁷⁹ and monopolize the market by shutting off supply to its customers.⁸⁸⁰ Based on the rationale of *Commercial Solvents* case law has developed the notion of the “essential facilities doctrine.”⁸⁸¹ The Commission argued in this regard that intellectual property is a property right like any other, so that its exercise can be constrained in the interest of protecting competition.⁸⁸² In U.S. case law thus far there is missing the affirmative application of the essential facility doctrine to cases involving intellectual property as an essential asset.⁸⁸³

competition with its long-standing customer. In response to the customer’s complaint the Commission found *Commercial Solvents*, in breach of Article 82 of the EC Treaty and based its reasoning on the evidence that *Commercial Solvents* intention after it had decided to integrate vertically, had been to drive its customer out of the market and to fill that gap in the market itself. *See Commercial Solvents*, para. 25 of the judgment.

⁸⁷⁸ Access to a downstream asset may give rise to a potential competition law concern when a firm active in both upstream and downstream activities refuses to provide access to the facility to other firms who wish to provide products that are purchased by end consumers. The question then becomes whether the competition authorities should insist that the first firm provides access. *See, e.g., BISHOP & WALKER, supra* note 851, para. 6.105, at 239.

⁸⁷⁹ Heidinger, *supra* note 843, at 23.

⁸⁸⁰ In the famous case *Oscar Bronner GmbH & Co. KG v. Mediaprint Zeitungs- und Zeitschriftenverlag GmbH & Co. KG and Ors*, Case C-7/97 [1998] ECR I-7791, [1999] 4 CMLR 112 [hereinafter *Bronner*], Advocate General Jacobs stressed the importance of focusing on harm to competition and therewith to consumers, not to competitors: “[I]t is important not to lose sight of the fact that the primary purpose of Article [82] is to prevent distortion of competition—and in particular to safeguard the interests of consumers—rather than to protect the position of particular competitors.” *Id.* at para. 58; *but cf.* the in a judgment rendered six years before *Bronner* the ECJ then pursued the competition policy that “[a] dominant undertaking which both owns or controls and itself uses an essential facility...and which refuses its competitors access to that facility...thereby placing the competitors at a competitive disadvantage, infringes Article [82].” *See Case IV/34.174 B&I Line plc v. Sealink Harbours Ltd. and Sealink Stena Ltd.* [1992] 5 CMLR 255, at para. 41.

⁸⁸¹ *See, e.g., Case C-27/76* [1978] ECR 207, [1978] 1 CMLR 429 *United Brands Company and United Brands Continentaal BV v. Commission*; *see also Schuhmacher, supra* note XX, at 93.

The use of the concept of an essential facility has had a significant and growing role in the Commission’s liberalization program being implemented in network industries such as telecommunications, gas, electricity and transport. However, the doctrine has also been invoked in other industries that are not generally considered to be characterized by natural monopoly. Such industries include newspaper distribution, port facilities and marketing data. For its most recent application in the *Microsoft* case as to interface implementation *see Part VII.2.2.5.*

⁸⁸² *See, e.g., ANDERMAN & KALLAUGHER, supra* note 870, at 285; *see also Schuhmacher, supra* note 850, at 106-8.

⁸⁸³ *Schuhmacher, supra* note 850, at 92 (mentioning *United States v. Microsoft Corp.*, 231 F. Supp.2d 144 (D.D.C. 2002)).

This, however, is not the case under European case law. The current position is that a licensor may be obliged to grant a license, but only under “exceptional circumstances”.⁸⁸⁴ What are the circumstances *in concreto* that may render a refusal to supply abusive? It has been with ECJ to emerge with an answer in pertinent case law.

2.5. Relevant Case Law

2.5.1. Magill

The first case applying the essential facility doctrine to copyright with the remedy of a compulsory licensing order imposed upon the copyright owner is the *Magill* case.⁸⁸⁵ Three major broadcasting companies were found to breach Article 82 for refusing to supply to an Irish publisher of a weekly news magazine (“Magill”) information on the programs they would be showing on their respective channels that was protected by copyright under Irish law. The television companies traditionally produced their own separate TV listings of the programs that they would be showing on their respective channels, but Magill wanted to produce its own comprehensive weekly TV guide on the Irish market that would give information on the programs that would be shown on all three channels.⁸⁸⁶ The ECJ held that the refusal to allow use of the full program information⁸⁸⁷ constituted an abuse under Article 82 of the EC Treaty. In its reasoning the ECJ affirmed the principle that the refusal of a dominant company holding an intellectual property right to grant a license cannot in itself constitute an abuse of a dominant position. However, the exercise of an exclusive right by a proprietor may, in exceptional

⁸⁸⁴ Cf. ULLRICH, *supra* note 76, at 77 (qualifying decompilation as not *per se* anti-competitive conduct simply because the program’s secrets will be uncovered, rather the purpose for decompiling another’s program is relevant and with keeping the “why” in mind copyright protection may not be (mis)used to an anti-competitive extent unless certain circumstances may justify that behavior).

⁸⁸⁵ Commission Decision (89/205/EEC) *Magill TV Guide: ITP, BBC and RTE v. Commission*, 1989 O.J. (L 78) 43 (decision dated Dec. 21, 1988); following a lengthy judgment from the CFI in 1991, Case T-69/89, *RTE v. Commission* [1991] ECR II-485, the case was finally resolved by the ECJ on Apr. 6, 1995 in Joined Cases C-241/91 P and C-242/91 P *Radio Telefis Eireann (RTE) and Independent Television Publications Ltd. (ITP) v. Commission* [1995] ECR I-743 [hereinafter *Magill*].

⁸⁸⁶ *Magill*, at para. 6 et seq. of the 1995 ECJ judgment.

⁸⁸⁷ Subject to a contractual limitation on its use *Magill* was allowed to print only weekly highlights in its news magazine but not complete listings.

circumstances, involve abusive conduct. Where were the circumstances in *Magill* that rendered the refusal to supply abusive? First, the television companies prevented, by their actions, the emergence of a new product for which there was “a potential consumer demand.”⁸⁸⁸ In the second place, there was no actual or potential substitute for the copyright from each of the broadcasters. Third, the ECJ found no objective justification for the refusal to supply.⁸⁸⁹ The fourth circumstance held to be crucial in this determination, was the finding that the television broadcasting companies had statutory (copyright) monopolies by virtue of their scheduling of TV programs in Ireland and the United Kingdom and as such they were trying to reverse for themselves the secondary market of TV guides. The ultimate aim of the broadcasters was to exclude all competition in the secondary market for TV guides “since they denied access to the basic information which is the raw material indispensable for the compilation of such a guide.”⁸⁹⁰

2.5.2. IMS Health

In 2004, the ECJ delivered another judgment ordering mandatory licensing in a case where an intellectual property right was involved.⁸⁹¹ In Germany, IMS Health GmbH & Co. OHG (“IMS”) was a provider of pharmaceutical regional sales data services based on the “1860 brick structure”. The structure was used to report the estimated sales of pharmaceutical products in each zone.⁸⁹² NDC Health GmbH & Co. KG (“NDC”), a competing supplier of these services, attempted to break into the German market. However, NDC failed to do so largely because potential customers had adapted their information and distribution systems to the “1860 structure”.⁸⁹³ Under German law NDC was barred from using the “1860 structure” since the structure was considered a database and as such protected under copyright

⁸⁸⁸ *Magill*, at para. 54 (referring to *Volvo*).

⁸⁸⁹ *Id.*, at para. 55.

⁸⁹⁰ *Id.*, at para. 55 (in relying on *Commercial Solvents* licensing of the listings was found to be an indispensable input (facility) for the comprehensive TV guide).

⁸⁹¹ Case C-418/01 *IMS Health GmbH & Co. OHG v. NDC Health GmbH & Co. KG* [2004] ECR I-5039 [hereinafter *IMS Health*].

⁸⁹² *IMS Health*, at para. 4.

⁸⁹³ *Id.*, at para. 5.

law. In effect, IMS held a dominant position at the relevant market. NDC's request for a license was refused by IMS and thereby IMS abused its dominant position.

In the judgment the ECJ confirmed its previous position that intellectual property owners are in general free to exercise their exclusive rights and that only in "exceptional circumstances" a refusal to license amounts to an abuse of a dominant position under EC competition law.⁸⁹⁴ The Court set out the following conditions for establishing where the refusal by a copyright owner to grant access by way of a license may be regarded as an abuse:⁸⁹⁵

[T]he refusal of access to the facility is likely to eliminate all competition in the relevant market; such refusal is not capable of being objectively justified; and the facility itself is indispensable of carrying on business, inasmuch as there is no actual or potential substitute in existence for that facility.⁸⁹⁶

The ECJ identified three criteria which have to be met cumulatively⁸⁹⁷ in order to find a conduct abusive:

- the refusal is such as to reserve the copyright owner the relevant market by eliminating all competition in that market,
- the refusal cannot be justified by objective considerations, and
- the undertaking that requests the license intends to offer new products or services not offered by the copyright owner and for which there is a potential consumer demand.

We have already met each of these criteria in the copyright context though under different headings. In Europe, decompilation to be permissible under copyright law is primarily aimed at promoting competition in the worldwide software market. The second prong covers the objectives defined by law to be justified as underlying Articles 5 and 6 of the EU Software Directive. The third prong is the equivalent to the Directive permitting decompilation for the purpose of creating new interfacing competing products. In terms of U.S. copyright law, the first prong equals the policy based approach of the copyright misuse doctrine. The second prong resembles the "legitimate reason" requirement as established under the *Sega* rule.

⁸⁹⁴ *Id.*, at para. 34.

⁸⁹⁵ *Id.*, at para. 37.

⁸⁹⁶ *Id.* at para. 70.

⁸⁹⁷ This remained unclear after the *Magill* judgment.

Finally, the *Sony* court would label the third prong the “transformation” requirement.

Notwithstanding these striking similarities of copyright and competition law, none of both jurisdictions so far has contemplated a compulsory licensing order as remedy in a pure copyright case⁸⁹⁸ where the copyright owner could determine what and to what extent interface information embedded in its copyrighted computer program would be necessary to be disclosed in order to fully satisfy the degree of interoperability as prescribed by law. Otherwise an “all-or-nothing” approach has been adopted in both copyright law regimes.

2.6. The Bifurcated Role of Disclosure under Copyright Law

Intellectual property law and competition law are two legal regimes that can and do conflict. In addition to the tension between competition and intellectual property law with respect to a monopolist’s right to engage in exclusionary practices,⁸⁹⁹ compulsory licensing orders in copyright cases encounter the tension within intellectual property law itself, between the need to provide exclusive rights so as to preserve the incentive to create, and the need to insure access so as to facilitate the creation of new intellectual products.⁹⁰⁰

The EU Software Directive states explicitly that copyright protection is not to be extended to ideas and principles⁹⁰¹ and, “to the extent that logic, algorithms and programming languages comprise ideas and principles”, those aspects of a computer program likewise lie outside the scope of copyright protection mandated by the Directive.⁹⁰² On the one end of the spectrum, disclosing (i.e., granting access to) a program’s techniques will entail not only ideas and principles but also expression. Requiring disclosure of the expression of the work goes beyond

⁸⁹⁸ This is not true for all forms of intellectual property. Under French patent law, for instance, the government may require the compulsory licensing of a patent, if among other things, within a specified period of time the patentee has failed to exploit the patent or to sell the product in which it is embodied in quantities sufficient to satisfy the demand of the French market. *See Cotter, supra* note 826, at 192 n.71. However, given the different policies of patent law and copyright law with respect to a disclosure requirement of the protected works there is need to caution against coming up with a pro-compulsory licensing argument based on a hasty analogy.

⁸⁹⁹ *See discussion supra* Part VII.2.1.

⁹⁰⁰ Cotter, *supra* note 826, at 179.

⁹⁰¹ EU Software Directive, *supra* note 10, art. 1(2), at 44.

⁹⁰² *Id.*, recital 14, at 43.

copyright's purpose and can actually prove to be counterproductive.⁹⁰³ The returns that a software programmer earns from investments affect the programmer's incentive to invest in development in the first place (R&D expenditures, above all, will be an issue).⁹⁰⁴ Without such returns, i.e., if the programmer's results of which can be appropriated by competitors through mandatory entry without similar (R&D) cost can counteract to reduce such investments with consequent adverse effects on dynamic market efficiency.⁹⁰⁵

On the other end of the spectrum, access might be the *quid pro quo* for the public's granting exclusive rights to the author. Intellectual property licensing generally will expand the use of protected works, increasing competition for their use, and disperse the technology thereby encouraging improvements.⁹⁰⁶ However, neither the Berne Convention⁹⁰⁷ and the Universal Copyright Convention⁹⁰⁸ nor the EU Software Directive and any of its preceding legislative drafting mentions disclosure as an objective of European copyright policy. Thus, no basis for a conceptual commitment to access as a *quid pro quo* for copyright protection can be discerned from existing laws in Member States, international copyright conventions, or the Directive itself.⁹⁰⁹ It shall rather lie with the original program writer himself to decide whether to publish his program's interface specifications or not and whether to go after infringing acts of uncovering such information or not.⁹¹⁰

⁹⁰³ For example, under German case law it has been established that software license agreements do not automatically imply an obligation of the licensor also to disclose the source code in addition to the object code of the computer program vis-à-vis the licensee. MARLY, *supra* note 111, at 268.

⁹⁰⁴ BISHOP & WALKER, *supra* note 851, para. 6.107, at 240 (stating that access to an essential facility, including intellectual property, must be confined to appropriate conditions, rather than providing broad grounds for regulatory interventions (e.g. compulsory licensing orders) which might undermine investment incentives).

⁹⁰⁵ Haaf, *supra* note 23, at 414; *see also* MORITZ & TYBUSSECK, *supra* note 16, at 67 (denying existence of an obligation under German copyright law whereby publication of scientific results would be the "entrance ticket" to the recognition of a monopoly-like copyright); *see also* BISHOP & WALKER, *supra* note 851, para. 6.107, at 240

⁹⁰⁶ Cf. GELLHORN, *supra* note 857, at 482.

⁹⁰⁷ Berne Convention, *supra* note 33.

⁹⁰⁸ Universal Copyright Convention, Sept. 6, 1952, 6. U.S.T. 2731, 216 U.N.T.S. 131 (as amended at Paris on July 24, 1971).

⁹⁰⁹ Haaf, *supra* note 23, at 415.

⁹¹⁰ *See* MORITZ & TYBUSSECK, *supra* note 16, at 67; *cf. also* *Data General Corp. v. Grumman Systems Support Corp.*, 36 F.3d 1147, 1187 (1st Cir. 1994) (holding that "while exclusionary conduct can include a monopolist's refusal to license a copyright, an author's desire to exclude others from use of its copyrighted work is a presumptively valid business justification for any immediate harm to consumers").

2.7. Microsoft

In the 21st century, computer software constitutes intellectual property of enormous financial value. Thus it is not surprisingly that cases involving intellectual property rights in computer software are most likely to be dealt with in a competition law setting. One such example is the revolutionary *Microsoft* case. Since the case began in 1998 the world has been waiting for the ruling by the European Court of First Instance (CFI) on a major aspect of the dispute: Microsoft's ability to protect its dominant Windows operating system without providing the information necessary for competitors to develop software compatible with Windows.⁹¹¹ For the purpose of this paper, the revolution possibly triggered by the *Microsoft* judgment is how European authorities have related the remedy of compulsory licensing with copyright aspects in a way as not seen before.

2.7.1. The Facts

On December 22, 2004, the CFI addressed for the first time the issue of software interoperability. It provisionally affirmed the European Commission decision relating to a proceeding under EU competition law in the *Microsoft* case.⁹¹²

The Commission's investigation into Microsoft began in 1998 when Sun Microsystems Inc. ("Sun") lodged a complaint with the Commission against Microsoft,⁹¹³ which holds a 95 per cent share of the worldwide market for PC operating systems with its Windows products.⁹¹⁴ According to Sun, Microsoft infringed Article 82 by refusing to license certain information⁹¹⁵ that Microsoft

⁹¹¹ Kevin J. O'Brien, *European Verdict on Appeal by Microsoft Expected Monday*, N.Y. TIMES, Sept. 17, 2007, available at http://www.nytimes.com/2007/09/17/technology/17soft.html?_r=2&oref=slogin&ref=technology&pagewanted=print&oref=slogin (last visited Sept. 17, 2007).

⁹¹² See Commission of the European Communities, Commission Decision of 24 March 2004 (Case COMP/C-3/37.792 Microsoft), available at <http://europa.eu.int/comm/competition/antitrust/cases/decisions/37792/en.pdf> (last visited March 28, 2007) [hereinafter Commission Decision in Microsoft].

⁹¹³ See *id.*, at 5.

⁹¹⁴ See *id.*, at 119 n.555.

⁹¹⁵ This information is not the Windows source code, which is not actually necessary to create interoperable products, but rather the "hooks" at the edge of the source code that allow communication between products. See News Release No. 45/04, European Commission, European Commission Concludes Microsoft Investigation, Imposes Conduct Remedies and a Fine (March 24,

contends is protected by intellectual property law.⁹¹⁶ According to Sun, this information is necessary in order for Sun to achieve full interoperability between its software products for network computing, called work group server operating systems,⁹¹⁷ and Microsoft Windows.⁹¹⁸ Sun contends that without access to this information that will allow it to achieve interoperability, Sun and other rivals will not be able to compete with Microsoft in the market for work group server operating systems.⁹¹⁹ After investigating for over five years, in March of 2004 the Commission concluded that Microsoft Corporation (“Microsoft”) had violated Article 82 of the EC Treaty by refusing to license to competitors information that would permit them to create software that is compatible with Microsoft’s dominant operating system for personal computers (“PC operating system”). The conduct found by the Commission⁹²⁰ to constitute an abuse consisted in Microsoft’s refusal to supply its competitors with interoperability information and to authorize them to use that information for the purpose of developing and distributing products competing with its own products on the work group server operating system market.

By way of remedy, the Commission decided in favor of Sun, ordering Microsoft, *inter alia*, to disclose to any undertaking wishing to develop and distribute work group server operating systems, within 120 days, those interface specifications required by them in order to enable their products to communicate with the

2004), available at <http://www.eurunion.org/news/press/2004/20040045.htm#%283%29> (last visited March 28, 2007) [hereinafter Commission Investigation in Microsoft].

⁹¹⁶ See *Commission Decision in Microsoft*, *supra* note 912, at 55.

⁹¹⁷ The Commission has described work group server operating systems as “operating systems running on central network computers that provide services to office workers around the world in their day-to-day work, such as filesharing, security and user identity management.” *Commission Investigation in Microsoft*, *supra* note 915.

⁹¹⁸ See *Commission Decision in Microsoft*, *supra* note 912, at 5.

⁹¹⁹ *Id.* (stating that Microsoft has acquired a dominant position in the market for work group server operating systems).

⁹²⁰ The second type of conduct to which the Commission took exception was the tying of Windows Media Player with the Windows PC operating system. The Commission considered that that practice affected competition on the media player market. By way of remedy, the Commission required Microsoft to offer for sale a version of Windows without Windows Media Player. The CFI also confirmed to require Microsoft to “unbundle” its Windows Media Player.

Windows operating system.⁹²¹ In addition, the Commission imposed on Microsoft a fine of EUR 497.2 million for abusing its market power in the EU.⁹²²

On June 7, 2007 Microsoft brought an action before the CFI for annulment of the decision or for annulment or a substantial reduction of the fine imposed on it.⁹²³

2.7.2. The Judgment

Today's statement ... marks an important milestone in the Commission's efforts to address Microsoft's continuing abuse of its dominant position and ensure competition on the merits in key software markets.

(Thomas C. Vinje, ECIS Spokesman and Legal Counsel)

On September 17, 2007, the CFI issued its long-awaited judgment. Ultimately, the CFI has not reduced the fine and confirmed, *inter alia*, the imposition of the Commission's remedies requiring the disclosure of interoperability information to competitors.⁹²⁴

The aim pursued by the Commission is to remove the obstacle for Microsoft's competitors represented by the insufficient degree of interoperability with the Windows domain architecture, in order to enable those competitors to offer work group server operating systems differing from Microsoft's on important parameters. In respect to that count, the CFI centers its ruling on the degree of interoperability.⁹²⁵ The court observes that the Commission defined interoperability information as a detailed technical description of certain rules of interconnection and interaction that can be used within Windows work group networks to deliver

⁹²¹ Microsoft will be entitled to "reasonable remuneration for any of this interface information which may be protected by intellectual property rights." See *Commission Decision in Microsoft*, *supra* note 912, at 299.

⁹²² This fine relates both to Microsoft's refusal to deal as well as its tying violation. While it is the largest fine that the Commission has ever imposed on a single company for violations of EC competition law, the Commission could have established an even larger fine. EC competition law permits the Commission to impose fines of up to 10 per cent of a firm's worldwide turnover for the preceding business year but the Commission calculated the fine based on Microsoft's European turnover only. See *id.*; see also Donna M. Gitter, *Strong Medicine for Competition Ills: The Judgment of the European Court of Justice in the IMS Health Action and Its Implications for Microsoft Corporation*, 15 DUKE J. OF COMP. & INT'L L. 153, 187 (2005).

⁹²³ Case T-201/04 [2005] ECR II-1491 (intervention); Case T-201/04 R [2004] ECR II-4463 (interim relief – Article 82 EC Treaty); Case T-201/04 R [2004] ECR II-2977 (interim relief – intervention).

⁹²⁴ Case T-201/04 *Microsoft Corp. v. Commission* [2007] ECR (not yet published) [hereinafter CFI Judgment].

⁹²⁵ *Cf. id.*, at para. 210.

work group services.⁹²⁶ The court then notes that the Commission emphasized that Microsoft's abusive refusal to supply concerned only specifications of certain protocols and not the source code⁹²⁷ and that it was not its intention to order Microsoft to disclose its source code to its competitors.⁹²⁸ In that connection, the CFI rejects Microsoft's claims that the degree of interoperability required by the Commission is intended in reality to enable competing work group server operating systems to function in every respect like a Windows system and, accordingly, to enable Microsoft's competitors to clone or reproduce its products.⁹²⁹

Regarding the refusal to supply the interoperability information, the court recalls that, according to the case law,⁹³⁰ although undertakings are, as a rule, free to choose their business partners, in certain circumstances a refusal to supply on the part of a dominant undertaking may constitute an abuse of a dominant position. The CFI reiterated that before a refusal by the holder of an intellectual property right to license a third party to use a product can be characterized as an abuse of a dominant position, three conditions must be satisfied under the essential facility doctrine: (i) the refusal must relate to a product or service indispensable to the exercise of an activity on a neighboring market; (ii) the refusal must be of such a kind as to exclude any effective competition on that market; and (iii) the refusal must prevent the appearance of a new product for which there is potential consumer demand. Provided that such circumstances are satisfied, the refusal to grant a license may constitute an abuse of a dominant position unless it is objectively justified.⁹³¹

The court considers that the work group server operating systems of Microsoft's competitors must be able to interoperate with Windows domain architecture on an equal footing with Windows operating systems if they are to be capable of being marketed viably. Since the word "interoperability" can be used in different contexts

⁹²⁶ *Id.*, at para. 195 et seq.

⁹²⁷ In technical terms, there is a distinction of the concept of "specifications" and the concept of "implementation". Whereas specification is a description of what a computer program must achieve in an abstract way, implementation relates to the actual computer code that will run on the computer. *Id.* at para. 199.

⁹²⁸ *Id.*, at para. 203-4.

⁹²⁹ *Id.*, at para. 212.

⁹³⁰ The CFI appraised the facts in *Microsoft* in the light of *Magill* and *IMS Health*. *See id.*, at para. 116.

⁹³¹ *Id.*

by technicians, the judgment⁹³² refers to the definition of interoperability as laid down in the recitals of the EU Software Directive.⁹³³ As to the second prong, the absence of such interoperability has the effect of reinforcing Microsoft's competitive position on the market and creates a risk that competition will be eliminated. The court observes that the circumstance relating to the appearance of a new product must be assessed under Article 82(b) of the EC Treaty.⁹³⁴ Microsoft's refusal limits technical development to the prejudice of consumers within the meaning of that provision.⁹³⁵ Last, the CFI rejects Microsoft's arguments to the effect that the refusal is objectively justified because the technology concerned is covered by intellectual property rights. The court further notes that such justification would render ineffective the principles established in the case law which are referred to above.⁹³⁶ The court considers that Microsoft has failed to show that if it were required to disclose the interoperability information that would have a significant negative effect on its incentives to innovate.⁹³⁷

After all, the CFI confirms that the necessary degree of interoperability required by the Commission is well founded and that there is no inconsistency between that degree of interoperability and the remedy imposed by the Commission.⁹³⁸

2.8. *Highlights of the Microsoft Judgment*

2.8.1. Shall Future Compatibility Be Protectible?

As a "highlight" of the Microsoft Judgment, the Commission's order required Microsoft to update its interface information as it brings to market new versions of

⁹³² *CFI Judgment, supra* note 924, at para. 156 et seq.

⁹³³ EU Software Directive, *supra* note 10, recitals 10, 11 and 12, at 43.

⁹³⁴ Article 82(b) of the EC Treaty, at , reads as follows:

"Any abuse by one or more undertakings of a dominant position within the common market or in a substantial part of it shall be prohibited as incompatible with the common market in so far as it may affect trade between Member States.

Such abuse may, in particular, consist in: ...

(b) limiting production, markets or technical development to the prejudice of consumers[.]"

⁹³⁵ *CFI Judgment, supra* note 924, at para. 648 et seq.

⁹³⁶ *Id.*, at para. 683.

⁹³⁷ *Id.*, at para. 697.

⁹³⁸ *See id.*, at para. 753.

its products.⁹³⁹ The Commission pointed out that Microsoft ultimately could upgrade its operating system so as to destroy the compatibility achieved by its rivals through their decompilation efforts.⁹⁴⁰ This particular part of the Microsoft Judgment appears to contradict with the intent of Article 6(1)(c) of the EU Software Directive. The scope of the lawful decompilation is limited to those parts of the target program which are necessary to be decompiled in order to achieve interoperability.⁹⁴¹ Neither the wording itself nor the legislative drafting provide reference to the temporal purview of the condition.

In the United States, the Federal Circuit in *Atari* dealt with the question as follows. Atari sought to copy Nintendo's entire 10NES security program to ensure compatibility with future versions of the Nintendo console. Atari feared that if it did not copy the entire 10NES program, Nintendo may update the next version of the console to monitor portions of the data stream that are currently unmonitored, allowing the console to once again distinguish between Nintendo and Atari games. Thus far the facts are congruent with Sun's line of argumentation in its complaint. The *Atari* court sided with Nintendo on this issue, holding that Atari may not copy portions of the program needed only for future compatibility. The court noted the importance of the balance in public policy between the dissemination of valuable ideas within an industry and the protection of a manufacturer's "lead time" in the market.⁹⁴² In other words, Atari has the right to adopt Nintendo's technology, but only after Nintendo has presented it to the marketplace. If Atari could copy present code for future compatibility, it could drastically reduce the time needed to reverse engineer Nintendo's next version of the 10NES. This rationale applied in the *Microsoft* case would mean that Microsoft is ordered to share its fruits of creativeness on a long-term basis with other market players.

2.8.2. The "Ultima Ratio" Approach Revisited: Compulsory Licensing

⁹³⁹ See *Commission Decision in Microsoft*, *supra* note 219, at 300.

⁹⁴⁰ See *id.*, at 184-5.

⁹⁴¹ See EU Software Directive, *supra* note 10, art. 6(1)(c), at 45.

⁹⁴² See *Atari*, 975 F.2d at 843.

The Directive does not rule on the remedies and sanctions for the infringement of software copyright.⁹⁴³ The EU legislators have rather left it with the Member States to adopt remedies which are in accordance with their national civil law and criminal law remedies.⁹⁴⁴ Generally, compulsory licensing is a very rare remedy.⁹⁴⁵ On several occasions, the European legislators allocated the remedy of mandatory licensing, thereby access to copyrighted material, to be applied under competition law rather than copyright law.⁹⁴⁶

Notwithstanding the lack of any codified or express mandate to share copyrighted material with others—save various exceptions relating to the notions of fair use and the interoperability in competitive terms—a compulsory licensing approach is at the wake of recognition and permission in case law to resolve the incentive/access tradeoff in the copyright context.⁹⁴⁷ Whereas, the United States are focusing on market incentives to compel the voluntary licensing of intellectual property in general,⁹⁴⁸ Europe has recently chosen to rely on the essential facility doctrine with compulsory licensing as its consequence. It appears not coincidental that the first Commission compulsory licensing decision concern products subject to a utilitarian copyright.

In *Microsoft*, the Commission suggests that Microsoft's refusal to license the information that would allow its competitors to achieve interoperability of their

⁹⁴³ However, the need and significance of harmonizing copyright remedies based on the TRIPS Agreement has been recognized at European level. See *Commission Report*, *supra* note 36, at 18.

⁹⁴⁴ For an overview of the remedies and sanctions regimes on a state-by-state basis see *Commission Green Paper*, *supra* note 47, para. 2.6.55 et seq., at 65 et seq.; see also ULLRICH, *supra* note 76, at 87-8.

⁹⁴⁵ The same is true for intellectual property rights enforced under U.S. law. See, e.g., *Dawson Chemical Co. v. Rohm & Haas Co.*, 448 U.S. 176 (1980) (“[c]ompulsory licensing is a rarity in our patent system”); see also *Orson, Inc. v. Miramax Film Corp.*, 189 F.3d 377 (3rd Cir. 1999) (ruling that the state may not mandate the distribution or reproduction of a copyrighted work); see also Cotter, *supra* note 826, at 191-2 (noting that the access-insuring practice of compulsory licensing thus far has largely been avoided in the United States—with only a handful of exceptions, for example, the federal government's power to compel a patentee or copyright owner to license his work to the government upon the payment of just compensation).

Similarly, the Berne Convention permits developing countries to require the compulsory licensing of the right to reproduce and translate copyrighted works for purposes of teaching, scholarship and research, if certain conditions are met. See Berne Convention, *supra* note 33, at Appendix arts. II, III.

⁹⁴⁶ Cf. Council Directive 92/100/EEC on rental right and lending right and on certain rights related to copyright in the field of intellectual property, O.J. (L 346) 61, art. 13 and recital 47.

⁹⁴⁷ See Cotter, *supra* note 826, at 201 (noting that the essential facility doctrine threatens to trump the intellectual property rule whereby there is no duty of an owner of an intellectual property to share her creation with others).

⁹⁴⁸ *Id.*, at 193.

software stems from its desire to monopolize the market for work group server operating systems.⁹⁴⁹ The Commission declared that the processes of decompilation and reverse engineering, respectively, were not a viable solution for Microsoft's competitors because the process would be too time-consuming in such a fast-moving industry and possibly unsuccessful due to the complexity of the information.⁹⁵⁰ Yet, it is neither clear from the EU Software Directive itself nor from the legislative history what happens if decompilation is not objectively indispensable but the would-be competitor is simply subjectively incapable of decompiling the target computer code so to accomplish interoperability. Compulsory licensing may be an answer.

Compulsory licensing proves to be an efficient and non-infringing alternative to decompilation of computer programs. Therefore, adopting the Commission's proposal, may likely bridge the decompilation dilemma, including the intermediate copying problem, in the legal community. If the European Communities' courts do not accept the Commission's reasoning in a copyright context, however, they might conclude that compulsory licensing is inappropriate in light of the ability of Microsoft's competitors to deduce the interoperability information they otherwise seek through decompilation.

In the United States, compulsory licensing as an alternative to reverse engineering, for the time being, has not been a substantial issue. Although evidence showed that the defendant Accolade might have been able to obtain the computer code by licensing it from Sega provided that Sega be the exclusive manufacturer of all games produced by Accolade, the Ninth Circuit considered that condition to access to be unreasonable without undergoing any further in-depth discussion.⁹⁵¹

At first blush, an equivalent defense to the common law doctrine of intellectual property misuse in general, and copyright misuse in particular, is lacking in European copyright law. However, the analysis undertaken by the European authorities in the *Microsoft* case resembles the practice U.S. courts would use had

⁹⁴⁹ *Commission Decision in Microsoft*, *supra* note 912, at 158-9 (stating that Microsoft had made its interoperability information available in order to gain a foothold in the market for work group server operating systems and thereupon withheld such information once its work group server operating systems gained acceptance).

⁹⁵⁰ *See id.*, at 183-4.

⁹⁵¹ *See Sega*, 977 F.2d at 1514.

the copyright misuse defense been invoked in *Microsoft* litigated in the United States. At the end of the day, yet the European approach appears to weigh in favor of open systems and even more depart from the U.S. outcome as the CFI has recently proven its willingness to order compulsory licensing in a software interoperability context.

3. *Compulsory Licensing of Intellectual Property Rights*

3.1. Compulsory Licensing As The Fairest Solution?

Some platform developers publish interfaces, some maintain their interface specifications as closely held trade secrets, and others license them—either more or less freely.⁹⁵² “Non-open software” inherently implies a refusal to share the program’s interface specifications. Decompiling a computer program, as one alternative to access unprotected elements of computer code, can be the key to the program’s “vault”. As a side-effect, decompilation may easily end up in uncovering also copyrightable expression or information which later on can be utilized in connection with competing products. Those unpleasant consequences can be avoided if the original software developer decides to license only the portion of the computer code required by the potential reverse engineer. Licensing then will satisfy the interests of both involved.

While the aim of competition law is to create efficient markets, intellectual property laws create monopolies for a limited period of time. The law of compulsory licensing therefore strikes a balance between the entitlement of intellectual property owners to exercise their exclusive rights and the need to maintain competition on markets.⁹⁵³

If it comes to interoperability concerns, the platform producer can ensure exclusivity of its platform by making exclusivity a condition of licensing.⁹⁵⁴

Notwithstanding the fact that a refusal to license is not generally be considered

⁹⁵² Samuelson & Scotchmer, *supra* note 5, at 1616.

⁹⁵³ Heidinger, *supra* note 843, at 23.

⁹⁵⁴ Both Sega and Nintendo tried to keep their interfaces proprietary by forbidding licensees to make games for other platforms. Sega initiated a lawsuit to stop Accolade from making games for its proprietary platform or adapting games made for other platforms. *See Sega*, 977 F.2d 1510; *Atari*, 975 F.2d 832.

anti-competitive in its nature licensing of computer code to software programmers to write application programs for a particular operating system may well be deemed anti-competitive, for example, if the licensor also requires those programmers not to write applications for competing operating systems.⁹⁵⁵ In addition to constraining the exercise of market power by dissuading other potential market entrants the licensor can recoup its R&D expenses with the license fee.⁹⁵⁶ For this end, the cost of reverse engineering must be high enough in order to permit the platform developer to charge an adequate license fee.⁹⁵⁷ When will the cost of reverse engineering high enough? Over many years Microsoft's operating system has become a *de facto* standard platform for applications running on personal computers. Simultaneously, Microsoft's interface specifications have become more complex and have been altered frequently. Without any willingness to enter a licensing agreement, for the reverse engineer it would cause even more money, time and other resources to decompile Microsoft's API in order to be capable of undertaking the process of decompilation. The only reason why it may occur now and then that application software developers prefer to decompile a platform's interface information and so waiving the easier and quicker way of licensing may be their will to escape from the reach of the platform developer's exclusivity terms included in the licensing agreement.⁹⁵⁸

Moreover, it can be the payment of a license fee in compulsory licensing cases that may reconcile the factual consequences of the decompilation exception with the objective of the EC Treaty to ensure an undistorted competition in the Community's internal market. Article 5(2) of the EC Treaty obligates the Member States faithfully to accomplish the objectives of the EU,⁹⁵⁹ particularly to strengthen the competitiveness of Community industry and to ensure undistorted

⁹⁵⁵ BISHOP & WALKER, *supra* note 851, paras. 5.61, 5.63, at 173, 174.

⁹⁵⁶ In the arbitration case of *IBM v. Fujitsu*, the arbitrators recognized a legitimate interest of the original program developer to participate in the exploitation of the secondcomer which the latter could only achieve due to the interface information embedded in the decompiled program. *See MORITZ & TYBUSSECK, supra* note 16, at 81.

⁹⁵⁷ *See* Samuelson & Scotchmer, *supra* note 5, at 1586.

⁹⁵⁸ Although it would have been easier and quicker to license the Sega Genesis interface, Accolade would have had to stop writing for other platforms, due to Sega's insistence on exclusivity. *See, e.g., Sega*, 977 F.2d at 1514.

⁹⁵⁹ EC Treaty, art. 5(1), at 42, in full text:

“Any action by the Community shall not go beyond what is necessary to achieve the objectives of this Treaty.”

competition in the Community's internal market.⁹⁶⁰ A compulsory licensing approach applied to cases involving decompilation matters could run the risk to interfere with these objectives as the approach would open the door to the target program's information necessary to achieve interoperability predominantly to big software companies.⁹⁶¹ While decompilation of computer software can almost only be afforded by big sized software companies, the lack of technical and financial resources of smaller sized software companies will establish an obstacle to factual access of interface specifications in the way bigger software companies do have. The crux is that licensing is usually done for a fee (also called royalty). It is just the license fee—even if calculated in a fair way and under the auspices of the courts—that financial restraints upon small and medium sized software companies, in reality, could result in a *de facto* one-sided support of big software companies that can afford the license fee to be paid in return for the information desired. Additionally, small and medium sized firms in the software industry normally are dependent on interface information of the standard software to be capable of creating their own compatible software. The dominant provider of the standard software can alter the interface code any time, at whim and caprice, thereby pressurizing smaller software companies whose success at the software market goes with the up-to-date compatibility of their products. Consequently, the practice of ever-changing interface information embedded in standard software likely will eliminate smaller software providers from the market as they do not have sufficient resources at their disposal to keep abreast of the current standard.⁹⁶² Those situations would not suffice to achieve the objective of strengthening the competitiveness of Europe's software industry. The competition in the internal market, consequently, would be distorted since free competition shall also include

⁹⁶⁰ Article 3(1) of the EC Treaty, at 40-1, in the pertinent parts reads as follows:

“[T]he activities of the Community shall include, as provided in this Treaty and in accordance with the timetable set out therein:...

(g) a system ensuring that competition in the internal market is not distorted;...

(m) the strengthening of the competitiveness of Community industry;

(n) the promotion of research and technological development;”

⁹⁶¹ *Cf. MARLY*, *supra* note 111, at 328.

⁹⁶² *Id.*, at 309-10 (indicating that non-standard software developer have hard times to economically survive the fast pacing world of the computer industry since they face the threat that their own created computer programs may be already outdated when released at the market because of the in the meantime changed interface code of the standard software whereby leaving the newly marketed application software without any (standard) platform to run on).

viable competition between small and medium sized software companies on the one side and big companies on the other side.⁹⁶³

It is one thing to mandate access (i.e. the grant to license an essential facility) yet another to establish the terms at which that access is granted.⁹⁶⁴ A commentator suggested to interpret the essential facility doctrine as to authorize courts only to forbid a dominant supplier from refusing to deal with a given customer or competitor, but not to regulate what those terms and conditions must be.⁹⁶⁵ In *IMS Health*, the Commission delegated the task of determining whether fees are reasonable in the given circumstances to the parties (or in this case, to IMS and the party requesting the license). Alternatively, in those circumstances where mutual agreement cannot be reached, the fees are to be set with reference to independent expert opinions.⁹⁶⁶ A contrary opinion points out that in the absence of price regulation, the owner of the essential facility, the copyright owner of the program including interface specifications, can effectively deny access by requesting too high a price.⁹⁶⁷ Furthermore, the argument that a compulsory licensing order will not subsidize smaller software companies goes awry if they must pay the prevailing rate as charged to the licensee's similarly situated competitors. However, if there are no other competitors as comparative measure, the rate should be determined in view to the reverse-engineering cost that would accrue had the licensee the resources to gain access through his own acts of decompilation.

The amount of the license fee should then be measured on the costs that unlicensed secondcomers would have to bear in decompiling the target program and how long it would take them.⁹⁶⁸ That standard for determining the license fee will ensure that small and medium sized software companies will be at the same footing as big sized companies. While bigger industry players with sufficient financial and technical resources anyway will be in the position to decompile the target software and subsequently reverse-engineer their own software, smaller industry players may seek a compulsory licensing order by court. In so doing, those smaller sized

⁹⁶³ See Article 3(1)(g) of the EC Treaty, at 40.

⁹⁶⁴ BISHOP & WALKER, *supra* note 851, para. 6.122, at 249.

⁹⁶⁵ Cotter, *supra* note 826, at 211, *cf. also Intergraph Corp. v. Intel Corp.*, 3 F.Supp.2d 1255 (N.D. Ala. 1998).

⁹⁶⁶ See *IMS Health*, at para. 17.

⁹⁶⁷ BISHOP & WALKER, *supra* note 826, para. 6.114, at 244.

⁹⁶⁸ *Cf. Samuelson & Scotchmer*, *supra* note 5, at 1589.

companies will get access to the information necessary to develop their own interconnecting software and therewith possibly entering the market as (more or less evenly matched) competitors to bigger software companies. European case law reflects the approach that mandating access to an essential facility is ordinarily to be coupled with price regulation—in a way as to ensure both non-discriminatory and reasonable⁹⁶⁹ license terms.⁹⁷⁰ After all, the question is how likely in practice small and medium sized software developers will be in fact capable to afford the license fee is written on another page.

Without doubt, small, innovative software developing companies stand a lot to gain from the judgment, compared with larger players that either have sufficient technical know how to overcome the computer code blockade anyway or sufficient financial resources to settle with Microsoft, like Sun. In any event, the judgment in *Microsoft* will level the playing field for these smaller, next-generation innovators who did not have these (interoperability) agreements before. Now that the playing field is level, however, the question is still whether their products will catch on and whether they can deliver that know how.

3.2. Microsoft's Perspective

How is the compulsory licensing order perceived by Microsoft? In the *Microsoft* case, licensing may well lead to an increase in demand of Microsoft's operating system software in the same way as IBM benefited from the wide array of application software developed for IBM's standard PC.⁹⁷¹ In furtherance of its dominant position on the software market Microsoft can avail itself of "network effects"⁹⁷² in two ways. First, the value of a program depends, to some extent, upon the number of other people who know how to use the program and therefore can share data employing it. For example, knowing how to use Microsoft Word is more valuable if many data files, in particular those from different entities, are written using Microsoft Word as opposed to other competing application programs.

⁹⁶⁹ See *IMS Health*, at para. 17.

⁹⁷⁰ BISHOP & WALKER, *supra* note 851, para. 6.115, at 245.

⁹⁷¹ Samuelson & Scotchmer, *supra* note 5, at 1616.

⁹⁷² See Cotter, *supra* note 826, at 202-3.

Second, the value of a program also depends in part upon the program's interoperability with complementary goods (such as application programs). The more people use the Microsoft proprietary platform, the more valuable it becomes because more application programs will be written to be compatible with the Microsoft operating system than with other systems.⁹⁷³ If these network effects are strong enough, both factors may reinforce the operating system's dominance, or the establishment of an industry standard, and at the same time squeeze rivals out of the marketplace.⁹⁷⁴ Then access to the dominant operating system will become essential to competition in the market for application programs. Here applying the essential facility doctrine will protect consumers from unfair pricing and ensure efficiency and competition in related, complementary markets.⁹⁷⁵ However, the wider the array of application software compatible with Windows, the more Microsoft's dominant position at the software market will become entrenched. Broadly licensing of Windows' interface specifications may not only be beneficial for consumers and competitors but on a long-term perspective also for Microsoft. From this point of view Microsoft should rethink its rigorous stance against the compulsory licensing order of the CFI.

After the CFI Judgment, Microsoft will have to make substantial changes in the way it supplies interoperability information to competitors seeking to have their work group server software work with Microsoft's Windows.⁹⁷⁶ Microsoft announced that it will provide open source software developers access to and use of its interoperability information, according to the CFI Judgment. That means that Microsoft will offer the interoperability information under license terms that will permit programmers to use it in open source software. Such software grants competitors unfettered rights to copy, modify and redistribute the software, freedoms that Microsoft's earlier terms would not permit. Microsoft thus fears to lose market share because of the openness of its interface architecture that may put

⁹⁷³ *Id.*, at 204.

⁹⁷⁴ Samuelson & Scotchmer, *supra* note 5, at 1618.

⁹⁷⁵ Cotter, *supra* note 826, at 204.

⁹⁷⁶ See Brad Smith, Response to European Court of First Instance Ruling: Remarks by Brad Smith, Senior Vice President, General Counsel, Corporate Secretary, Legal & Corporate Affairs of Microsoft, Brussels, Sept. 17, 2007, available at <http://www.microsoft.com/Presspass/exec/bradsmith/09-17-07CFI.msp> (last visited Nov. 20, 2007).

other operating system developers in the position to clone Windows. Consequently, Microsoft has agreed not to enforce its patents over interoperability information only for non-commercial open source software projects. However, if the software is used for commercial purposes, Microsoft may likely try to assert its exclusive rights under intellectual property law.⁹⁷⁷

⁹⁷⁷ See Microsoft Statement Regarding the European Court of First Instance's Judgment in the EU/Microsoft Case, Statement by Microsoft Corp. Senior Vice President and General Counsel Brad Smith, Luxembourg, Sept. 17, 2007, available at <http://www.microsoft.com/presspass/press/2007/sep07/09-17Statement.mspx> (last visited Nov. 20, 2007).

VIII. Conclusion—A Current Status Report

In the 1950s, the computer industry in the United States has been targeting the general public, rather than the individual author of a computer program: a communitarian view of software development was the norm, and programmers openly shared their ideas with one another. As the technological *zeitgeist* took its pace, not only in the United States but also in Europe, U.S. software developers recognized the vast profitability of software, and they began to exercise tight controls over their computer codes by different means. The United States have shifted toward a closed proprietary model of software development. As a common law tradition nation, federal jurisdiction in the United States tried to settle the decompilation battle via fair use tactic. Despite the fact that the balancing test of the fair use factors under section 107 of the Copyright Act leaves some discretion to U.S. judges, the parties involved in the decompilation dilemma are caught in a “no win” situation in either way: On the one end of the spectrum, an extension of the fair use doctrine into the area of decompilation would leave software providers powerless to stop the use of their software by others, especially competitors, to plunder profits that would otherwise belong to the software providers. At the other end of the spectrum, without the extension of the fair use doctrine into the area of decompilation, software providers would be able to receive copyright-like protection of functional elements embodied in their software due to the fact that the utilitarian aspects of the software are not directly observable.

The European Communities have verged to establish an identity in the global software world, especially to be able to form a counterpart to the U.S. dominance in the software market. These efforts culminated in 1991 in the enactment of the EU Software Directive with the highly controversial decompilation exception of Article 6. By adopting a reasonable devised exception for decompiling computer programs Europe has come up with a worldwide unique⁹⁷⁸ model of how to walk the thin line between public and private interests in the reverse engineering

⁹⁷⁸ As the first piece of legislation on copyright protection of computer programs the EC Software Directive of 1991 has had ramifications on the development of copyright protection regarding computer software in different ways, particularly in the United States and on the legislation adopted by the WIPO. *See* Vinje, *supra* note 32, at 251.

software industry. The decompilation exception may be cited as manifest for Europe's position in the decompilation battle, as one towards "open systems". Article 6 centers around interoperability as the indispensable criterion to lawful software decompilation under European copyright law.

The CFI recently stated that the degree of interoperability and the scope of the remedy are "intrinsically linked".⁹⁷⁹ Interoperability is a double-edged sword in the decompilation battle. On the one hand, interoperability serves as legitimate reason for decompiling computer software permissible under copyright law. On the other hand, software interoperability functions to ensure competition in a market dominated by a monopolist, like an owner of copyright in its computer program. Thus, interoperability is the key criterion in both copyright law and competition law involving software decompilation. Although the objectives pursued under copyright and competition law are different, the remedy of compulsory licensing of essential facilities (such as interface specifications as basis for accomplishing interoperability) may be efficient to the same extent in either law. A mandatory licensing order in pure copyright cases would effect to bridge the inherent conflict of software copyright and decompilation in a way as it would be unique in the world copyright law regimes. However, thus far it has been an issue exclusively for European competition law to determine to which extent competition law can be used to force compulsory licensing of intellectual property rights.⁹⁸⁰ Beyond the traditional role of the compulsory licensing remedy in competition law, mandatory licensing could also serve as remedy to reconcile the contradicting private and public interests in the software decompilation context.

In *Microsoft*, a case settled in European competition law, the CFI aimed at reconciling Microsoft's interest in its intellectual property and the general public's interest in intellectual growth, which is one of the objectives of the EU Software Directive, by way of compulsory licensing beyond the traditional scope of copyright law. The same imbalance has occurred under U.S. copyright law, though solved in a different way. The Ninth Circuit, in *Sega* and *Sony*, stripped of the copyright incentives and rewards that first had coaxed software providers into producing and disseminating their works, while thereby preserving competition.

⁹⁷⁹ CFI Judgment, *supra* note 924, para. 152.

⁹⁸⁰ ANDERMAN & KALLAUGHER, *supra* note 870, para. 10.33, at 280.

Harsh consequences, like in *Sega* and *Sony*, might have been avoided by imposing on them the obligation to enter into reasonable licensing agreements with Accolade and Connectix. Generally, the current frustration among U.S. software providers could so be mitigated in two respects: First, by way of compulsory licensing Sega and Accolade would be financially reimbursed for their creative efforts, skill and resources spent in software programming. Second, the practice of compulsory licensing clearly expresses the recognition of the software providers' copyright and their protection thereunder. Both these reasons may contribute, on the one hand, to keeping the innovative software industry alive and growing, and, on the other hand, to strengthening the confidence of software providers in the copyright protection of their computer software. Nonetheless, compulsory licensing as an alternative to decompilation in the United States, for the time being, has not been a substantial issue. In response to the competitive implications of software decompilation U.S. courts cautiously apply the copyright misuse defense.

In the face of *Magill*, *IMS Health* and *Microsoft*, there is a perception that the Commission considers that the protection afforded by national intellectual property rights is prone to go beyond the legitimate reward for the software author's creative and financial efforts. It is in cases involving intellectual property rights where there is the greatest danger of misuse of an essential facility.⁹⁸¹ While U.S. software providers may invoke copyright misuse defense, European case law—devoid of a copyright misuse doctrine or alike—refers to the essential facility doctrine for software providers to apply in cases where national copyright laws provide for near-monopoly protection up to an anti-competitive extent.⁹⁸² Under European law, it is competition law that may oblige a dominant owner of the essential facility, including computer code, to cooperate with its competitors, on competition grounds. In so doing, courts have relied on basic principles of antitrust economics of the impact of refusal to deal.⁹⁸³ A similar balancing test, which is yet neither undisputed nor uniformly applied by U.S. judges, has been established under U.S.

⁹⁸¹ BISHOP & WALKER, *supra* note 851, para. 6.118, at 246.

⁹⁸² *But see* Cotter, *supra* note 826, at 180 (recognizing, on the one hand, justifications for an expanded right of access to intellectual property, while, on the other hand, criticizing the essential facility doctrine to be, because of its “hazy and uncertain contours”, an inappropriate legal device to realize that right).

⁹⁸³ BISHOP & WALKER, *supra* note 851, para. 6.123, at 250 (citing the former senior Commission official Mr. Temple Lang in 1994).

case law when referring to the doctrine of copyright misuse. To day, the misuse defense has not been codified in the Copyright Act. Neither has the U.S. Supreme Court yet clarified the existence and scope of copyright misuse. This reluctance indicates another concession to enhanced protection of software copyright owner vis-à-vis reverse engineers, and thereby continues to pursue a “closed source code” model.

At the other end of the Atlantic, the European compulsory license approach as adopted in *Microsoft*, from a copyright standpoint, proves efficient likewise in that compulsory licensing of interface specifications for the purpose of interoperability may contribute, on the one hand, to keeping the innovative software industry alive and growing, and, on the other hand, to strengthening the confidence of software developers in the copyright protection of their computer software. So the unwanted side-effects triggered off by “all-or-nothing” decisions, such as in *Sega* and *Sony*, could then be avoided in future. In affirming the Commission’s decision in the *Microsoft* case the CFI appears to reinforce Europe’s position in the software industry, particularly vis-à-vis dominant U.S. software providers. By responding to the refusal of access to interface information contained in copyrighted computer software with a compulsory licensing order the European Communities would give further direction to an “open software” model.

After all, this paper advocates the Commission’s position that compulsory licensing would be the most “open” solution in the decompilation context because it would allow persons who engage in software decompilation unrestricted access to another program’s interface specifications upon the payment of a compulsory fee as reimbursement for the software provider’s efforts in the development of that interface. Furthermore, compulsory licensing in copyright cases under the auspices of the judiciary would likely deter reverse engineers from deliberately seeking access to copyrighted computer code for purposes of subsequent software piracy. However, this paper questions whether the United States will follow the *Microsoft* approach given the U.S. copyright law as evolving at different settings into a “closed software” attitude. With the transformative approach adopted under *Sony* the permissibility of decompilation (and reverse engineering) has been limited to the creation of only non-competing programs. *Reimerdes*, a case decided under the

DMCA, leads to an ambit of the reverse engineering exemption narrower than the fair use doctrine because the court in finding copyright infringement focused on the early stage of intermediate copying in the overall process of reverse engineering. Finally, there may be observed a trend under contract law according to which anti-decompilation (and anti-reverse engineering) clauses are deemed enforceable with the effect that minimum decompilation rights may be contracted away in private stipulations by the U.S. software author.

One thing is certain: changing technology will continue challenging the creativeness of lawyers, legislators and judges as well as software providers to adapt the law, and copyright law in particular, to situations never contemplated by existing statutes or previous case law in the United States just as in Europe. This paper suggests one way of how the decompilation battle could be waged with less blood spilling. Given the diverging interests of the involved in an ever-pacing technological high-speed environment it, arguably, would be illusory to believe in an entire and final dispute settlement. Certainly this paper once will belong to legal history and maybe I will re-write it starting with: “Introduction—May I Introduce Jason’s Mom and Her Grandchild”.

Copyright © 2008 by Petra Heindl