SOFTWARE REVERSE ENGINEERING AMENDMENTS IN SINGAPORE AND AUSTRALIA

Jonathan Band[1]

During the Instant Messaging battle this past summer, Microsoft acknowledged that it had reversed engineered America Online's protocols in order to achieve interoperability with AOL's mail system. Saul Hansell, *In Cyberspace, Rivals Skirmish Over Messaging*, New York Times, July 24, 1999 at A1. Although the precise technical details were not made public, this incident demonstrates the continuing relevance of software reverse engineering to competition in the networked digital environment. And since the Internet is borderless, the next significant Internet product may be developed in and distributed from Sydney or Singapore rather than San Jose -- making the legal status of software reverse engineering outside the United States more important than ever to the future of the information technology industry.

In the past year and a half, both Singapore and Australia have amended their copyright laws to permit software reverse engineering. The two countries employed different legislative processes and statutory approaches, yet ended up in similar places for similar reasons. Singapore quietly developed an approach that closely follows the U.S. reliance on the fair use doctrine. In contrast, the Australians pursued a lengthy and often contentious deliberative process to arrive at an amendment modeled on the European Union's 1991 Software Directive. In both instances, the governments made clear that the amendments were necessary to allow their domestic software industries to compete in the global market.[2]

---

[1] Jonathan Band is a partner in the Intellectual Property Group of the Washington, D.C. office of Morrison & Foerster.

[2] Computer programs typically are distributed in object code -- machine readable zeros and ones. A developer of an interoperable program can discern the rules of interconnection - the interface specifications - only by reverse engineering this object code. One method involves translating, or "decompiling," the machine readable object code into a higher level, human readable format. Some have argued that this translation for research purposes violates the first programmer's copyrights, even if the final interoperable product released to the market is unquestionably non-infringing.

Singapore

Singapore's amendment was necessitated by a 1996 ruling of the Singapore Court of Appeal in *Creative Technology Ltd. v. Aztech Systems Pte Ltd,* Civ. App. No. 181 at 1995 (November 12, 1996).[3] In *Aztech*, the Court of Appeals reversed the trial court's holding that Aztech's copying of Creative Technology's program during the course of reverse engineering it to develop a compatible product was a "fair dealing" under the Singapore Copyright Act (SCA).

Section 35(1) of the SCA provided that "fair dealing…for the purpose of research or private study shall not constitute an infringement of the copyright." Section 35(5) defined "research" as excluding "industrial research, research carried out by bodies corporate…or bodies or persons carrying on any business." The trial court had ruled that Aztech's reverse engineering was private study and not research, and therefore permitted under Section 35(1). *Creative Technology* at ¶¶75-77. The Court of Appeal, however, opined that this broad interpretation of private study would render Section 35(5) meaningless, and concluded that private study could not be performed for commercial purposes.

The Court of Appeal's decision had the affect of prohibiting software companies from engaging in reverse engineering in Singapore. In response, the Attorney-General of Law drafted an amendment to the Copyright Act, which was made public when it was introduced in the Singapore Parliament in February, 1998. Copyright (Amendment) Bill of 1998. The amendment deleted Section 35(5), thereby allowing a court to interpret research and private study to include commercial reverse engineering. In introducing the amendment, the Attorney-General of Law stated "the deletion … of section 35(5) of the Act will bring us in line with the United States, the United Kingdom, other European Union countries, Hong Kong, and Australia, which

[3] For a more extensive discussion of this case, *see* Jonathan Band and Taro Isshiki, *Interoperability in the Pacific Rim: Reversal of Fortunes in Singapore and Australia*, Journal of Proprietary Rights at 2 (July 1997).

do not bar the use of copyright materials for commercial research."[4] Second Reading of Copyright (Amendment) Bill of 1998 (February 19, 1998).

Professor Chin Tet Yung, in the brief debate of the amendment in Parliament, said:

> It is very important to ensure that there is a fair balance in any Copyright Bill between the interests of holders of rights in "cutting edge" software and the interest of competitors who want to design and market non-infringing competing programmes which interface or are inter-operable with the basic programmes.
>
> The Court of Appeal's decision in Creative Technology v. Aztech established that currently Singapore's copyright law does not permit most kinds of reverse engineering. Companies cannot decompile programmes to establish how they were put together and armed with that knowledge to develop new inter-operable programmes. Whether competitors should be able to reverse engineer and, if so, to what extent, is a very difficult matter to resolve. It seems clear, however, that most countries in the world are trying to draw a line between those two differing computer industry groups so that those who own the copyright in the leading programmes can maintain their strong copyright protection over their software, but that in certain circumstances others may decompile because there is a public interest in doing so.
>
> In the United States, use is made of the "fair use" defence, whereby courts are required to weigh up, on the facts of every case, whether the defendants could justify their activities. In Singapore, with the current amendment to section 35(5), I am pleased to see that the Copyright Bill brings the law of Singapore very close to that of the United States. This is especially welcome and should receive warm support from the industry. *Id*.

[4] Before the turnover to China, Hong Kong amended its fair dealing provision to more closely follow the U.S. fair use exception. The stated purpose of the amendment was to permit software reverse engineering. *See* Jonathan Band, *Gunboat Diplomacy on the Pearl River: The Tortuous History of the Software Reverse Engineering Provisions of Hong Kong's New Copyright Bill*, The Computer Lawyer at 8 (February 1998).

In short, the amendment was clearly intended to overturn the result in *Aztech* and permit software reverse engineering to the extent permitted by the U.S. fair use doctrine.[5] The government sought to allow Singapore companies to develop interoperable software products.

Australia

In contrast to Singapore's expeditious legislative process, Australia debated the issue of software reverse engineering for over a decade. In 1988, the Copyright Law Review Committee, an officially convened group of jurists, intellectual property lawyers, and industry representatives, began considering whether a software reverse engineering exception was needed. After holding public hearings and soliciting several rounds of comments from industry, the CLRC in 1995 issued a detailed report recommending adoption of a reverse engineering exception similar to that found in the EU Software Directive.[6]

In the ensuing four years, Australian government officials received extensive input on the CLRC recommendations from the domestic and foreign industry, as well as from the U.S. government. Finally, in the Spring of 1999, the government introduced in Parliament a set of copyright amendments relating to computer programs, including a reverse engineering exception, similar to the EU Software Directive. Copyright Amendment (Computer Programs) Act, No. 105 of 1999. The amendments passed the Senate on June 29, 1999, and the House of

---

[5] The U.S. Court of Appeals for the Ninth Circuit in *Sega Enter. Ltd. v. Accolade*, Inc., 977 F.2d 1510, 1527-28 (9th Circuit 1992), held that "where disassembly 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, disassembly is a fair use of the copyrighted work, as a matter of law."

[6] For a detailed discussion of the process leading up to the issuance of the CLRC Report, *see* Jonathan Band and Masanobu Katoh, *Interoperability Down Under: The Australian Copyright Law Review Committee's Final Repor*t, The Computer Lawyer at 20 (July 1995). For a detailed discussion of the EU Software Directive, *see* Jonathan Band and Masanobu Katoh, Interfaces on Trial: *Intellectual Property and Interoperability in the Global Software Industry* (Westview 1995) at 227-62.

Representatives on August 12, 1999. It was proclaimed and went into effect on September 30, 1999.

1. The Need For A Reverse Engineering Exception

In the Second Reading Speech on August 11, 1999, the Attorney-General, the Hon. Daryl Williams QC, explained the government's rationale for introducing the legislation. The Attorney-General described the growing importance of computers and computer networks to the economy. With the advent of the Internet, " there is an obvious need for computers and the programs which drive them to communicate, connect, or 'interoperate' with each other." Speech on Second Reading at 2. The Attorney-General then explained the need for interface information in order to achieve interoperability, and how this information as a technical matter can often be obtained only through reverse engineering. The Attorney-General singled out the reverse engineering technique known as "decompilation," which involves translating the machine readable object code into a higher level, human readable format. *Id*. at 2-3.

The Attorney-General noted that "the law of the leading software producing country in the world, the United States, allows makers of new programs to use decompilation to find out the interface information of existing programs for achieving interoperability. The countries of the European Union, and other countries, also allow this to be done. However, Australian law does not make such a provision." *Id*. at 3.

The Attorney-General contended that an amendment was required to enable the Australian software industry to compete in the world market.

> Australia's software producers are recognized as innovative by world standards. Because our industry is not of a scale to compete across the board with such dominant industries as that of the United States, its comparative advantage lies in the ability to cater for niche markets. In order to do this, it must be able to ensure that its successful niche products

> interoperate with other, existing products, including those produced by big scale producers….If Australian industry is to be allowed to compete on level terms with producers of similar products in the USA and Europe, Australian software copyright laws must be brought more into line with the law in those countries. *Id*. at 3-4.

At this point, the Attorney-General explained the provisions of the amendment: "as an exception to the copyright reproduction right, where interface information about other programs is not readily available to a software producer, the producer will now be able to decompile another program to the extent necessary to get the required interface information for making an interoperable product." *Id*. at 4. The Attorney-General hastened to add that the amendment would "not weaken the existing proscription of software piracy," explaining that pirates do not reverse engineer but engage in wholesale copying. *Id*.

Finally, the Attorney-General described two other reverse engineering exceptions created by the amendment: one for error correction, such as Y2K remediation, where an error free version is not available at a commercial price; and another for security testing, such as testing a computer's systems protection against hackers or viruses. *Id*. at 5-6.

2. The Structure of the Reverse Engineering Exception

In an Explanatory Memorandum which accompanied the amendment, the government discussed the four alternatives the government had considered. Explanatory Memorandum on Copyright Amendment (Computer Programs) Bill of 1999 at § 4.3. The first alternative was to leave the law unchanged. This alternative was rejected for the reasons outlined in the Speech on Second Reading: the costs to the Australian software industry would be too great.

The next alternative was to expand the fair dealing provisions of the Copyright Act, presumably to bring it more in line with the fair use provisions of the U.S. Copyright Act. Although this option was supported by the U.S. government and the Business Software Alliance,

the Australian government did not pursue this alternative because of the uncertainty to software developers concerning how much protection against infringement actions such a provision would provide. Since fair dealing, like fair use, is determined on a case by case basis by courts, the contours of the new fair dealing provision would emerge only from lengthy and expensive litigation. *Id*.

The third alternative was to adopt the reverse engineering provisions of the EU Software Directive. The government found this preferable to simply amending the fair dealing provision because a statutory exception provided more certainty to interoperable developers. At the same time, the government concluded that the Software Directive was deficient in two respects. First, it did not permit decompilation for purposes of security testing. Second, the Software Directive could be understood to permit decompilation only for purposes of achieving interoperability between two software products, but not between software and hardware.[7] The Australian government decided that decompilation should clearly be permitted for both software to software and software to hardware interoperability. Accordingly, the Australian government decided to pursue a fourth alternative: starting with the Software Directive, and adding provisions concerning security testing and software/hardware interoperability. *Id*.

The amendment passed by the Parliament has five sections concerning reverse engineering.

a. Black Box Reverse Engineering. Section 47B(3) parallels Article 5(3) of the Software Directive, and permits the copying done in the course of "black box" reverse engineering such as input-output tests. The section permits reproductions "made in the course of

[7] A manufacturer of a peripheral device such as a disk drive or a printer may need to reverse engineer a computer's operating system in order to ensure that the peripheral device properly functions with the computer. *See Interfaces on Trial* at 248-49 for arguments why the Software Directive permits decompilation for software to hardware interoperability.

running a copy of the program for the purpose of studying the ideas behind the program and the way in which it functions…."

b. Decompilation for Interoperability. Section 47D parallels Article 6 of the Software Directive, and permits making adaptations of a program (*e.g.*, decompiling a program) "for the purpose of obtaining information necessary to enable the owner or licensee to make independently another program (the new program), or an article, to connect to and be used together with, or otherwise to interoperate with, the original program or any other program." The reference to the making of "an article" is the language which permits decompilation for the purpose of achieving software to hardware interoperability. The wording of the final clause of the provision -- "to connect to and be used together with, or otherwise to interoperate with, the original program or any other program" -- makes clear that the exception is directed to the making of both products which attach to the original program and products which compete with the original program.[8]

The other provisions of Section 47D place limits on disassembly. Under subsection(c), the adaptation can be "made only to the extent reasonably necessary to obtain" the interface information. Under subsection (e), disassembly can be performed only when the interface information "is not readily available to the owner or licensee from another source when the … adaptation is made." Article 6 of the Software Directive contains similar limitations.

Section 47D contains a significant provision not found in the Software Directive; subsection(d) provides that "to the extent that the new program reproduces or adapts the original

[8] The attaching/competing question has long been a central issue in the reverse engineering debate. Dominant software vendors have argued that reverse engineering should be permitted only for the development of attaching, but not competing, products. However, because these dominant software vendors are typically vertically integrated, the attaching/competing distinction is artificial. For example, a new word processing product designed to "attach" to Microsoft Windows would also "compete" with Microsoft Word. Moreover, to achieve true backwards and forwards compatibility -- to make sure that the competitive product can interoperate with products on the market as well as those not yet introduced -- the competition often needs to examine both sides of the interface.

program, it does so only to the extent necessary to enable the new program to connect to and be used together with, or otherwise to interoperate with, the original program or the other program…." This subsection makes unambiguous that the interoperable developer can include in the new program the interface information derived from the original program. Although this concept is implicit in the Software Directive -- what would be the point of permitting disassembly if one could not use the fruit of that research? -- there is no explicit statement allowing the use of the information or declaring such information *per se* unprotected by copyright. The closest the Directive gets is in Article 1(2), which states that "[i]deas and principles which underlie any element of a computer program, including those which underlie its interfaces, are not protected by copyright…."[9] Section 47D(d) eliminates any ambiguity by directly permitting the copying of any element necessary for interoperability.

c. Error Correction. Section 47E permits reproducing or adapting computer programs to correct an error in the program that prevents it from operating as intended by its author or in accordance with specifications or documentation supplied with the original copy. The adaptations can be made only for the owner or licensee of a lawful copy of the original program; only to the extent reasonably necessary to correct the error; and only if an error free copy is not available within a reasonable time at a commercial price.

The Software Directive does not contain a detailed provision dealing exclusively with error correction. However, Article 5(1) of the Directive states that a lawful acquirer of a computer program may engage in any of the acts restricted by Articles 4(a) and (b) (including reproduction and translation) "where they are necessary for the use of the computer program … in accordance with its intended purpose, *including for error correction*." The Australian

[9] Under U.S. law, the unprotectability of interface information is based on judicial interpretation of 17 U.S.C. 102(b). *See, e.g., Computer Associates v. Altai*, 982 F.2d 693 (2d Cir. 1992); *Interfaces on Trial* at 83-165.

amendment, therefore, supplies additional specificity to a concept appearing in the Software Directive. This specificity appears to narrow the privileges granted under the Directive. Under Section 47E the error correction can be performed only if an error free copy is not available at an ordinary commercial price. Conversely, Article 5(1) of the Directive contains no such condition.

d. Security Testing. Section 47F permits the making of a reproduction or adaptation of a program for the purpose of 1) testing the security of the program, or a computer system of which the program is a part; or 2) investigating or correcting a security flaw or vulnerability in the program or a computer system of which the program is a part. This exception applies only if the information resulting from the reproduction or adaptation is not readily available from another source.

The Software Directive does not contain a parallel provision, but the dangers posed by hacking and viruses are better understood now than in 1991 when the EU adopted the Software Directive. Moreover, the U.S. Digital Millennium Copyright Act exempts computer system security testing from its ban on circumvention and circumvention devices. See 17 U.S.C. 1201 (j).

e. Limitation on Contractual Terms. Section 47H provides that "[a]n agreement, or a provision of an agreement, that excludes or limits, or has the effect of excluding or limiting, the operation" of the reverse engineering subsections, i.e., Sections 47B(3), 47D, 47E, or 47F, "has no effect." This provision prevents a software company from restricting the reverse engineering permitted under the amendment by imposing contract terms prohibiting such reverse engineering. The Australian government recognized that enforcing contractual restrictions on reverse engineering would undermine the pro-competitive and pro-interoperability objective of the legislation. The Software Directive contains a similar provision in Article 9(1).

Conclusion

Both Singapore and Australia have enacted copyright amendments intended to permit the reverse engineering necessary to achieve interoperability. Singapore opted to emulate the flexible case by case fair use approach of the United States. In contrast, Australia chose to follow the more certain civil code approach of the EU Software Directive. Despite the differences in the approaches, the same basic facts animated both governments: their software industries depend on interoperability; and interoperability often can be achieved only through reverse engineering. These basic facts impelled both governments to eliminate the legal barriers to software reverse engineering.

Interestingly, neither country appears to have considered the approach adopted by their neighbor, the Republic of the Philippines. The Philippines in 1997 crafted a hybrid of the U.S. fair use and EU Software Directive approaches by inserting the following sentence derived from the Software Directive in a fair use provision modeled on 17 U.S.C. Section 107: "Decompilation, which is . . . the reproduction of the code and translation of the forms of the computer to achieve the inter-operability of an independently created computer program with other programs may also constitute a fair use." Republic Act No. 8293 at §195. (Phil.)