

## 5 Interoperability Overseas

The Internet has convinced policy makers around the world of the importance of interoperability. In particular, Australian and Asian policy makers have concluded that their domestic firms can participate in the global market for information technology only if those firms' products can interoperate with the products developed by the dominant U.S. firms. As policy makers in the Asia-Pacific region have studied the issue, they have learned that the domestic firms can achieve interoperability only if they can reverse engineer the dominant firms' products.

Because of the nature of computer programs, most forms of software reverse engineering require the making of an interim copy of the program. Making such a copy may infringe the copyrights of the program's developer, unless it is permitted by an exception to the developer's exclusive rights. Two models for such exceptions emerged during the 1990s. First, the European Union adopted a software directive that contains a specific exception for reverse engineering.<sup>1</sup> (All the member states of the European Union have implemented the EU Software Directive, as have several other countries in Eastern Europe.) Second, courts in the United States found the copying incidental to software reverse engineering to be excused under the U.S. Copyright Act's fair-use doctrine.<sup>2</sup>

The Asia-Pacific countries considering the issue of reverse engineering had these two models before them. These countries, however, did not confront reverse engineering in a complete vacuum. First, the British Commonwealth countries had to consider whether their fair-dealing provisions,

1. Council 91/250/EEC 1991 O.J. (L 122). For more detailed discussions of the EU Software Directive, see section 1.2 above and pp. 227–282 of Band and Katoh, *Interfaces on Trial*.

2. We discuss U.S. case law concerning reverse engineering in subsection 1.3.2 and in section 2.2.

which were based on British copyright law, were flexible enough to permit software reverse engineering. The Court of Appeal in Singapore, for example, determined that its fair-dealing provision *did not* permit reverse engineering, which led the Singapore Parliament to amend the copyright law. Second, these countries had to deal with political pressure from dominant U.S. software companies and from the Office of the U.S. Trade Representative (USTR). These U.S. interests generally opposed any amendment permitting reverse engineering. At the same time, these U.S. interests signaled a preference for the U.S. fair-use approach over the EU Software Directive approach if the country decided to proceed with an amendment. Although the United States persuaded Hong Kong to adopt the fair-use approach rather than the Software Directive approach, it did not succeed in convincing it to abandon a reverse-engineering provision altogether, as it had previously convinced Korea and Japan.<sup>3</sup> Australia adopted the Software Directive approach notwithstanding strong U.S. government opposition. The Philippines responded to the U.S. pressure by enacting a hybrid of Software Directive and fair use.

It is not entirely clear why the USTR and the U.S. software firms preferred the fair-use approach to the Software Directive approach. As will be discussed below, the former is more flexible than the latter, and might permit a wider range of reverse-engineering activities. This preference for the fair-use approach might have reflected a belief that *Sega Enters. Ltd, Inc. v. Accolade, Inc.*<sup>4</sup> was an anomalous decision that U.S. courts eventually would reject. However, the opposite occurred; as was discussed in chapter 2, *Sega* has become more firmly entrenched in U.S. jurisprudence.

## 5.1 Australia

### 5.1.1 The Report of the Australian Copyright Law Review Committee

On April 12, 1995, the Australian Copyright Law Review Committee (CLRC) concluded a nearly eight-year study of copyright issues having to do with software.<sup>5</sup> The CLRC's 350-page final report culminated an open process of public hearings, several rounds of comments, technical demonstrations,

3. On the U.S. defeat of the Japanese reverse-engineering initiative, see pp. 297–316 of Band and Katoh, *Interfaces on Trial*. We discuss the defeat of the Korean initiative in section 5.5 of the present volume.

4. 977 F.2d 1510 (9th Cir. 1992).

5. This subsection is based on passages in the following publications: Jonathan Band and Masanobu Katoh, "Interoperability Down Under: The Australian Copyright Law

and draft recommendations.<sup>6</sup> For each of the many issues it considered, the final report carefully discussed all perspectives; it then reached the conclusion that a statutory amendment was necessary.

Among the more contentious issues to emerge in the course of the CLRC's deliberations were the protectability of interface specifications and the permissibility of software reverse engineering. As was noted above, these same issues contemporaneously were the subject of debate in the European Union and the United States. The CLRC explained its approach to these issues as follows:

[I]n the creation and protection of any property rights, an attempt must be made to strike the right balance between adequate protection and the need to provide the community with reasonable access to intellectual property and the benefits which it confers. . . . The striking of the balance is something which must be attempted in the public interest. The task has not been an easy one.<sup>7</sup>

In this subsection, we discuss how the CLRC approached its difficult task and ultimately succeeded in striking the right balance.

#### 5.1.1.1 The Copyright Law Review Committee

The Copyright Amendment Act of 1984 brought computer programs under the protection of the Australian copyright law. In October of 1988, the Attorney General asked the CLRC (an officially convened group of jurists, intellectual-property lawyers, and industry representatives) to consider whether Australian copyright law “adequately and appropriately protects computer programs.”<sup>8</sup> In February of 1989, the CLRC requested comments from interested parties on this question, and in April of 1990 it released an issues paper based on those comments. The paper recited the arguments for and against a reverse-engineering exception. Although the paper did not specifically address the protectability of interface specifications, it did discuss protection for program structure. Making no recommendations on the issues it raised, the paper invited further comments.

---

Review Committee's Final Report,” *The Computer Lawyer*, July 1995, at 20; Jonathan Band and Taro Isshiki, “Interoperability in the Pacific Rim: Reversal of Fortunes in Singapore and Australia,” *Journal of Proprietary Rights*, July 1997, at 2; Jonathan Band, “Software Reverse Engineering Amendments in Singapore and Australia,” *Journal of Internet Law*, January 2000, at 17.

6. Copyright Law Review Committee, *Computer Software Protection* (1995) (henceforth cited as CLRC Report).

7. CLRC Report at 4.

8. CLRC Issues Paper 1 (April 1990).

On July 26 and 27, 1990, the CLRC held a public hearing. Alcatel STC testified that reverse engineering was essential to the computer industry, and that copyright law should not be permitted to impede the practice. The U.S. Software Publishers Association, in contrast, testified that most copyright owners are vehemently opposed to permitting the reverse engineering of their products.

In September of 1990, IBM submitted written comments to the CLRC stating that “no special category of ‘fair use,’ or similar exception, [should] be created which might sanction the incidental copying of computer programs where it is part of the process of decompilation.”<sup>9</sup> In October of 1990, Fujitsu Australia submitted detailed comments taking an opposing view. Fujitsu Australia argued that if Australian software vendors “cannot develop products that conform to *de facto* interfacing standards (almost always established *outside* Australia, most typically in the U.S.),” then “programmers in Australia face extremely limited market opportunities (both in Australia and overseas).”<sup>10</sup> Accordingly, “the rules, formats, languages, protocols and similar information underlying a program, including its interfaces, should not themselves be copyrightable.”<sup>11</sup>

Fujitsu Australia further argued that “[r]everse analysis is an essential tool in the development of interoperable products. Unless discrete information can be discerned through machine analysis techniques, the development of compatible products can be frustrated; whether intentionally or otherwise. The absence of an ability to engage in reverse analysis would lead to *de facto* protection for product-to-product interfaces whenever a company failed to document that information.”<sup>12</sup> For this reason, Fujitsu Australia urged that reverse engineering “used to analyze or understand the uncopyrightable elements of a computer program” should be “viewed as completely permissible under Australian law.”<sup>13</sup>

On November 22, 1990, IBM conducted a demonstration of reverse engineering before the CLRC. IBM sought to show how decompilation facilitated disguised piracy. IBM also submitted a detailed paper on decompilation. IBM stated unambiguously that no reverse-engineering technique short of

9. IBM Submission to the CLRC 25 (September 1990). Decompilation is a technique of software reverse engineering that involves converting machine-readable object code into a higher-level, human-readable form.

10. Fujitsu Australia, Submission to the CLRC 11 (October 5, 1990).

11. *Id.* at 3.

12. *Id.* at 18.

13. *Id.* at 3.

decompilation is “regarded as objectionable or fundamentally inconsistent with the principles of copyright law.”<sup>14</sup> Nonetheless, IBM asserted, “there is no justification for legitimizing” decompilation, which “involves a flagrant infraction of the copyright owner’s exclusive rights to control reproduction, especially where the intent of such process is to quickly develop a substitute program.”<sup>15</sup>

IBM emphasized that decompilation would facilitate undetectable piracy, and that decompilation is unnecessary because “software suppliers publish ample information about their programs” and because less intrusive means of reverse engineering exist.<sup>16</sup> IBM argued that permitting decompilation would reduce a competitor’s costs and the first developer’s lead time. This, in turn, would reduce the incentives to create new programs. IBM, then the world’s largest computer vendor, was particularly solicitous of small developers: “An exception could impose particular hardship on small vendors who may have only a single successful innovative product.”<sup>17</sup>

Several developers of interoperable software sought to attend IBM’s reverse-engineering demonstration, but IBM insisted that the CLRC exclude them. This galvanized the developers of interoperable software into action. They formed Supporters of Interoperable Systems in Australia (SISA), which conducted its own reverse-engineering demonstration before the CLRC on February 7, 1991.<sup>18</sup> SISA also provided the CLRC with presentations on the users’ perspective and the business perspective. The main thrust of SISA’s advocacy was that the CLRC should clearly exclude interface specifications from copyright protection and should permit access to interface specifications by reverse engineering so that the Australian information-technology industry could compete effectively in a global market.

#### 5.1.1.2 *Autodesk v. Dyason* (1992)

On February 12, 1992, while the CLRC was considering the submissions of SISA and the proprietary vendors, the High Court of Australia handed down its decision in *Autodesk v. Dyason*.<sup>19</sup> This confused decision underscored the

14. IBM, Submission to the CLRC 6 (November 22, 1990).

15. *Id.* at 6–7.

16. *Id.* at 13.

17. *Id.* at 14.

18. SISA’s members included, in addition to several software companies based in Australia, the Australian subsidiaries of Bull, ICL, Unisys, NCR, Sun Microsystems, and Fujitsu.

19. *Autodesk Inc. v. Dyason*, (1992) 173 C.L.R. 330.

importance of the CLRC's work by demonstrating the need for clarification of the proper application of copyright law to computer programs.

Autodesk produced the computer-assisted design program AutoCAD. It sold AutoCAD with a hardwired lock that had to be physically installed on the PC or terminal on which AutoCAD was running. Because only one lock was sold with each AutoCAD program, the lock ensured that only as many copies of AutoCAD as had been purchased were in operation at any time. The AutoCAD program issued a repeating cycle of challenges to the AutoCAD lock. Using a shift register, the lock transmitted a series of responses to the AutoCAD program. The AutoCAD program then used a lookup table to determine whether the responses matched the challenges. If it did not receive a proper response, the AutoCAD program (not the lock) would issue instructions to stop the program.

After reverse engineering the AutoCAD lock, the defendants developed an "Auto Key lock" that responded properly to the challenges issued by AutoCAD. This made it possible to use copies of AutoCAD on PCs without the AutoCAD lock. The Auto Key lock, therefore, allowed a user to circumvent the Autodesk lock and use unauthorized copies of Autodesk.

The *Autodesk* case presented several issues of first impression for Australian courts. First, the trial court had to determine whether the hardwired AutoCAD lock constituted a computer program at all. The AutoCAD lock did not issue instructions; rather, it issued a data stream in response to the AutoCAD program's data stream, which the program then evaluated. Second, the court had to determine whether the defendant had copied any of the plaintiff's protected expression.

The trial court found for Autodesk, ruling that the AutoCAD lock was a computer program and that the Auto Key lock infringed the AutoCAD copyright because it performed the same function. The trial court's decision also implied that the defendant's studying the AutoCAD lock's output using an oscilloscope might be improper under copyright law.

The full Federal Court reversed the trial court, finding that the AutoCAD lock was not a computer program and that similarity in function did not infringe copyright. The High Court then reversed the Federal Court and restored the trial court's finding of infringement, but for reasons different from those articulated by the trial court. The High Court agreed with the Federal Court that the AutoCAD lock was a piece of hardware and not a computer program, and thus was not covered by the Australian copyright law. The High Court nonetheless found infringement because the defendant's Auto Key lock, even though a computer program, reproduced the protected expression of the AutoCAD lookup table, and this lookup table represented a "substantial part" of the AutoCAD program. Further, one of

the concurring opinions suggested, as did the trial court, that the defendant's act of reverse engineering also infringed Autodesk's copyright.

In a detailed criticism of the *Autodesk* decision, submitted to the CLRC, SISA stated that “[t]he danger of the High Court's *Autodesk* decision is that it will be portrayed as holding that all interface information by which two separate products interoperate is necessarily protectable ‘expression’ in Australia.”<sup>20</sup> SISA first argued that the High Court had erred in treating the lookup table as protected expression: “[d]ata which serves a purely functional purpose should not be viewed as expressive material.” SISA next argued that even if the lookup table contained expressive elements, those elements merged with the “idea” of achieving interoperability with the program:

[I]n order for Auto Key to work in place of the AutoCAD lock, the responses or return codes to signals sent to [the AutoCAD program] had to be identical. There was a very real functional imperative which limited the possible responses to stimuli sent from [the AutoCAD program] . . . [I]f a discrete amount of data that is passed between computer programs must be identical in order for the programs to work together, that very real absence of choice on the part of the developer creating a new product intended to work with or substitute for an existing product must be considered in determining whether, as to that bit series, “idea” and “expression” have merged.<sup>21</sup>

SISA further contended that *Autodesk* should not be interpreted as prohibiting reverse engineering. Finally, SISA argued that *Autodesk* was an instance of bad facts leading to bad law.

The High Court understandably had little sympathy for the defendant, who had sought to circumvent Autodesk's copy-protection device. Dyason's product was not intended to increase consumer choice, but to facilitate unauthorized copying. SISA concluded that “[i]n light of the High Court's *Autodesk* decision SISA believes more than ever in the importance” of amending Australian copyright law” to exclude interface specifications from protection and to permit reverse engineering.<sup>22</sup>

### 5.1.1.3 The CLRC's Draft Report

In July of 1993, the CLRC issued a 350-page draft report<sup>23</sup> that contained a lengthy and somewhat confusing discussion of U.S. case law on the scope of protection for non-literal program elements. The draft report recommended

20. SISA's Views on the High Court's *Autodesk* Decision 9–10 (1992).

21. *Id.* at 16.

22. *Id.* at 18.

23. CLRC Draft Report on Computer Software Protection (1993).

against any amendment to the Australian Copyright Act specifically dealing with the scope of protection. Thus, it did not support SISA's request for a provision specifically excluding interface specifications from copyright protection.

The draft report also contained a lengthy discussion of reverse engineering. It acknowledged that because programs typically are distributed in an object-code form that is not understandable by humans, certain exceptions to the copyright owner's exclusive rights are required to ensure that the public has access to the unprotected elements of the program. The draft report recommended the adoption of a provision similar to article 6 of the EU Software Directive, permitting decompilation for purposes of achieving interoperability.<sup>24</sup> Indeed, the draft report improved on the language of the Software Directive by eliminating article 6's confusing reference to the Berne Convention.<sup>25</sup> The draft report also endorsed permitting decompilation for error correction. Finally, the draft report recommended that the "fair dealing" provision of the Australian Act govern the permissibility of decompilation "to understand techniques."

Thus, the CLRC proposed decompilation rights somewhat broader than those under the Software Directive. The CLRC envisioned an unambiguous right to decompile for purposes of interoperability and error correction, and a flexible case-by-case fair-use approach to determining the lawfulness of decompilation to understand elements of the target program not related to interoperability. The draft report, however, made no mention of a provision similar to article 5(3) of the Software Directive, which permits black-box reverse engineering.<sup>26</sup>

24. The draft report recommended the following language:

[D]ecompilation of a computer program should be allowed where it is necessary to achieve the interoperability of an independently created computer program with other programs provided:

- (a) decompilation is performed by the owner of a lawfully acquired copy of the program or another person having a right to use the copy or on their behalf by a person authorized to do so;
- (b) the information necessary to achieve interoperability has not previously been readily available; and
- (c) the acts are confined to the parts of the program necessary to achieve interoperability.

The following limitations should apply:

- (i) the decompilation should only be used to achieve interoperability; and
- (ii) the information obtained should only be given to others when necessary for the interoperability of the independently created program.

25. See Band and Katoh, *Interfaces on Trial* at 254–255.

26. Black-box reverse engineering includes research methods other than decompilation, including line traces and input-output tests.



In detailed comments on the draft report, SISA stated that it believed that the CLRC's draft report struck a fair balance between the interests of copyright holders and the public at large. SISA also believed that the draft report was consistent with and in furtherance of the emerging international consensus of protecting computer programs as literary works while avoiding excess protection by creating express exceptions to the copyright holder's exclusive rights.<sup>27</sup>

SISA did, however, recommend several revisions. First, it renewed its request for a specific exclusion for interface specifications. Second, it advocated the adoption of a black-box reverse-engineering provision similar to article 5(3) of the Software Directive. Third, it recommended several changes to the equivalent of the Software Directive's article 6, most notably broadening it to cover explicitly decompilation for purposes of achieving interoperability between hardware and software.<sup>28</sup>

The U.S. Computer and Business Equipment Manufacturers Association (CBEMA) also filed comments on the draft report on behalf of proprietary vendors.<sup>29</sup> It offered several reasons why Australian copyright law did not need to permit decompilation for purposes of achieving interoperability. First, the policy objective of fostering interoperability "was already being satisfactorily advanced by a combination of market forces and liberal cross licensing policies of software developers," and "there is no evidence at all that a 'crisis' has developed in the industry due to an absence of special rules aimed at fostering interoperability."<sup>30</sup> Second, CBEMA hauled out the disguised-piracy rationale: "the protected expressions obtained through decompilation of computer program, often in an easily disguised form, can be used for a number of illegitimate purposes, which may cause substantial harm to the right holder."<sup>31</sup> Third, citing the reliance of the U.S. Court of Appeals for the Ninth Circuit on the fair-use doctrine to excuse

27. Comments on the Draft Report of the Copyright Law Review Committee on Computer Software Protection submitted by the Supporters of Interoperable Systems in Australia 1 (1993).

28. This had been a contentious, and somewhat unresolved, issue during the legislative battle leading up to the adoption of the EU Software Directive. See Band and Katoh, *Interfaces on Trial* at 248.

29. CBEMA's members included IBM, Apple, and Digital Equipment Corporation. In 2001, the organization changed its name to the Information Technology Industry Council.

30. CBEMA Comments on CLRC Draft Report 6 (October 1993) (hereafter cited as CBEMA Comments).

31. *Id.* at 7.

decompilation in *Sega v. Accolade*,<sup>32</sup> CBEMA argued that fair dealing (the Australian analogue to fair use) provided the means for “an Australian court to balance interests in the area.”<sup>33</sup> Further, “there is little evidence to suggest that Australian courts, as their U.S. counterparts, will have any difficulty in reaching fair results in specific factual situations through the application of the doctrine of fair dealing.”<sup>34</sup> The disingenuousness of this argument requires emphasis. In two *amicus* briefs to the Ninth Circuit, CBEMA vigorously opposed excusing decompilation under the fair-use doctrine. Yet in Australia CBEMA employed the Ninth Circuit’s fair-use finding as a justification for not adopting a specific decompilation exception.

Perhaps in response to SISA’s renewed request for a specific exclusion for interface specifications, CBEMA raised several arguments against the adoption of such an exception. CBEMA first argued that including specific terms of art in statutes “narrows the scope of the law, and, thus reduces its applicability in the full range of factual situations which may arise over time.”<sup>35</sup> CBEMA next argued that “the copyrightability of specific elements of a program should be determined by the same rules as the copyrightability of specific elements of any other work.”<sup>36</sup> CBEMA finally argued that an express exception for interface specifications for the purpose of promoting standardization and interoperability was unnecessary because over the past decade the software market had become increasingly standardized on its own: “[C]ompatibility (or interoperability) has evolved without specific rules diminishing protection.”<sup>37</sup>

The U.S. government also commented on the CLRC draft report. It approved the decompilation provision, noting that it “appear[s] to be generally consistent with the provisions of Article 6 of the EC Software Directive and appear[s] to be directed to achieving the goal of the creation of interoperable programs while protecting the copyright owner against abuse.”<sup>38</sup> The U.S. government, however, suggested that the decompilation provision explicitly state, as does Article 6 of the directive, that the “end result of the process must be the creation of a program that is itself

32. 977 F.2d 1510 (9th Cir. 1992).

33. CBEMA Comments at 6.

34. *Id.*

35. *Id.* at 3.

36. *Id.*

37. *Id.* at 4.

38. United States Government Comments on the Copyright Law Review Committee’s Draft Report on Computer Software Protection, United States Government Cable to United States Embassy, Australia (1993).

original.”<sup>39</sup> Moreover, the U.S. government expressed concern about permitting decompilation for error correction.

#### 5.1.1.4 The CLRC’s Final Report

The CLRC’s final report, issued in April of 1995, included several significant changes that favored interoperability.

It adopted the distinction, made by Professor Randall Davis of the Massachusetts Institute of Technology, between computer programs as “text” and computer programs as “behavior.”<sup>40</sup> The CLRC concluded that behavior should not receive copyright protection, thereby permitting the development of functionally equivalent programs with different texts.<sup>41</sup>

It did not accept SISA’s invitation to specifically exclude interface specifications from copyright protection. It did, however, explicitly endorse the *Computer Associates v. Altai* decision of the U.S. Court of Appeals for the Second Circuit.<sup>42</sup> It described the Second Circuit’s three-step test, noting that the unprotectable elements to be removed in the filtration step included “elements dictated by external factors, such as . . . compatibility requirements with other programs.” It then stated that the CLRC “regards the test set out by the court in that case as a very practical and useful guide for determining infringement if computer programs and supports the approach it adopted.”<sup>43</sup>

The report proceeded to discuss the favorable reception of *Computer Associates* by courts in Canada and the United Kingdom<sup>44</sup> and to reject the district court’s decision in *Lotus v. Paperback*.<sup>45</sup> (The appeal in *Lotus v. Borland* was still pending when the final report was drafted.) It also discussed “look and feel,” concluding that “the need for standardization and the need for efficient user interfaces to be used and developed outweighs the need to grant authors express copyright protection in the ‘look and feel’ of their programs’ behaviours.”<sup>46</sup> “While industrial efficiency may not be a consideration in determining protection of other categories of works,” the report stated, “that it is in the case of computer program serves to mark

39. *Id.*

40. CLRC Report at 102.

41. See *id.* at 112–113.

42. 982 F.2d 693 (2d Cir. 1992). See subsection 1.3.2 of the present volume.

43. CLRC Report at 109.

44. *Id.* at 109. See Band and Katoh, *Interfaces on Trial* at 147–149 and 262–269 for a discussion of these cases.

45. 740 F. Supp. 37 (D. Mass. 1990). We discuss this case in subsection 2.1.1.

46. CLRC Report at 114.

them out on account of their functional nature.”<sup>47</sup> These statements, taken together, suggest that the CLRC opposed copyright protection for interface specifications.

With respect to reverse engineering, the CLRC adopted SISA’s recommendation of a black-box reverse-engineering exception similar to that in article 5(3) of the Software Directive.<sup>48</sup> The CLRC also considered, and rejected, CBEMA’s opposition to a decompilation exception. The report concluded that the existing fair-dealing provision in Australian copyright law was narrower than the fair-use doctrine in U.S. copyright law and probably would not permit decompilation in a commercial context.<sup>49</sup>

The CLRC also rejected IBM’s contention that decompilation facilitated “disguised piracy.” It stated that it found IBM’s reverse-engineering demonstration unconvincing: “At that presentation, only a simple form of decompilation was demonstrated, namely the disassembly of a relatively small program. No evidence of generalised decompilation to high level computer languages was provided.”<sup>50</sup>

The CLRC noted that “any new program that is produced by reverse engineering an existing program and which is a copy or adaptation of the latter program is no less an infringement.”<sup>51</sup> The CLRC thus suggested that the focus of the copyright analysis should be the finished product brought to market, and not the intermediate development steps. The CLRC acknowledged IBM’s argument that many operating system interfaces were published, but responded that many other products were not publicly documented. The CLRC further observed that reverse engineering is time consuming, costly, and rarely leads back to a complete version of original source code. For this reason it is likely to be performed only by interoperable-software developers with no alternative means of obtaining the interface information necessary for interoperability.

The CLRC then considered SISA’s specific recommendations for amending the decompilation language proposed in the draft report. SISA had opposed the provision limiting decompilation to those parts of the program necessary for interoperability on the ground that it was not possible to know in advance what parts of the program needed to be decompiled. The CLRC agreed and replaced the problematic language. The CLRC also

47. Id. at 113.

48. Id. at 175–176.

49. Id. at 147–149.

50. Id. at 153.

51. Id. at 153.

agreed with SISA's suggestion that decompilation be permitted for purposes of achieving interoperability between hardware and software, as well as between two programs.<sup>52</sup> Further, the CLRC endorsed SISA's suggestion that contractual restrictions on reverse engineering not be enforceable. The CLRC reviewed the criticisms of its provision permitting decompilation for purposes of error correction, particularly those submitted by the U.S. government. The CLRC rejected the criticisms and retained the provision. The CLRC also retained its approval of decompilation to uncover ideas not related to interoperability pursuant to Australia's fair-dealing exception.

Finally, the CLRC considered the status of reverse engineering in other jurisdictions. It correctly concluded that its recommendations were consistent with the Software Directive in the European Union and the *Sega* decision in the United States, and that they complied with the Berne Convention and the World Trade Organization Agreement on the Trade Related Aspects of Intellectual Property Rights (TRIPS).<sup>53</sup>

Because of the parallels between the CLRC's recommendations and the EU Software Directive, the report is particularly useful in interpreting and applying the directive's provisions. Specifically, the report resolves three potential ambiguities in article 6 of the directive: it eliminates the confusing reference to the Berne Convention, it permits decompilation to achieve interoperability between software and hardware, and it removes the technologically infeasible limitation of decompilation to only those parts of

52. The final report worded the decompilation exception as follows:

[D]ecompilation of a computer program should be allowed where it is necessary to achieve the interoperability of an independently created computer program or hardware device with other programs or hardware devices provided. . .

- (a) decompilation is performed by the owner of lawfully acquired copy of the program or another person having a right to use the copy or on their behalf by a person authorized to do so; and
- (b) the information necessary to achieve interoperability has not previously been readily available; and
- (c) the acts are confined to those necessary to achieve interoperability.

The following limitations should apply:

- (i) the decompilation should only be used to achieve interoperability; and
- (ii) the information obtained should only be given to others when necessary for the interoperability of the independently created computer program or hardware device.

Id. at 10.

53. The final report states that "in Japan the law, if literally interpreted, would prohibit the reproduction and adaptation of computer programs for the purpose of reverse engineering" (CLRC Report at 177). A close examination of Japanese copyright law, however, reveals that it permits reverse engineering. See Band and Katoh, *Interfaces on Trial* at 294–297.

the program that are necessary for interoperability. The CLRC's thoughtful resolution of these potential ambiguities should be followed in Europe.

### 5.1.2 Case Law after the CLRC Report: *Data Access v. Powerflex* (1996)

After the CLRC issued its final report, an Australian federal court decided a case involving the protectability of functional components of a computer program dictated by compatibility concerns. The court harmed the cause of interoperability by holding that compatibility concerns could not negate the protectability of program elements. What makes this decision particularly interesting is its use and interpretation of United States copyright law, carefully selecting which U.S. decisions it agreed with.

#### 5.1.2.1 Facts

The *Powerflex* case focused on infringement by the end product, not infringement due to copying during the development process. The Data Access Corporation had developed DataFlex, a compilation of programs described by the Australian Federal Court "as an application development system."<sup>54</sup> Specifically, the DataFlex programs were used for not only the creation of databases but also for the development of database application programs, which could in turn be used to create and work with databases. As such, the DataFlex system provided users with a programming language in which they could write their own programs. The DataFlex system also incorporated a "run time" program that allowed a user to operate the applications created through the use of the DataFlex language. The defendants in the copyright infringement suit were David Bennett, his wife, and the company they had incorporated to sell the allegedly infringing product, PFXplus. Bennett, according to the court, "aspired to create an application development system which would be highly compatible with the DataFlex computer language."<sup>55</sup>

As far as issues of interoperability were concerned, the infringement suit focused on two aspects of Bennett's software. First, the PFXplus language used 192 of the 225 instruction words in the DataFlex language. The use of the same words in the source code of either the DataFlex or PFXplus languages caused the computer to perform the same functions. The program code implementing each of these functions, however, was completely different. The second interoperability issue arose in the context of Bennett's intentional reproduction, in identical form, of certain compression tables

54. *Data Access Corporation v. Powerflex Services Pty, Ltd.* No. 93-VG473 (Federal Ct. Austl. February 9, 1996) at 2.

55. *Powerflex* at 3.

used in one of the DataFlex programs. These tables, which were elements of the DataFlex “run time” program, merely allowed the software user to save storage space through the compression of the program into smaller data strings. This raised an interoperability issue because of Bennett’s assertion in his defense that it was necessary to use the same compression tables in his programs in order to achieve compatibility.

### 5.1.2.2 The Federal Court’s Ruling

#### *Merger of Idea and Expression*

Bennett presented a merger defense to the copying of the words of the DataFlex language. His merger defense was based on language in the U.S. Supreme Court decision *Baker v. Selden*,<sup>56</sup> relied upon by the Australian High Court in *Autodesk v. Dyason*,<sup>57</sup> that when an expression of an idea is inseparable from its function, it is part of the idea itself and therefore unprotected by copyright. Specifically, Bennett appealed to the logic of the First Circuit’s *Lotus Development Corp. v. Borland International*<sup>58</sup> decision (although not specifically to the “method of operation” argument, since the Australian copyright statute has no companion to section 102(b) of the U.S. Act). It appears that, as in *Borland*, Bennett had to use the same words in order to make PFXplus compatible with DataFlex and desirable to customers who had learned the DataFlex language and did not want to learn a new set of commands.

The Australian court chose to disregard the ruling in *Borland* and instead turned to Judge Keeton’s district court decision in *Lotus Development Corp. v. Paperback Software International*,<sup>59</sup> which as a practical matter had been overruled by the First Circuit’s *Borland* decision. The Australian court quoted extensively from the *Paperback* decision and seemed to place great importance on Judge Keeton’s decision that if the expression of an idea goes beyond the functional elements within that idea and beyond the obvious, *and* if there are numerous other ways of expressing the same idea, that form of expression is copyrightable.<sup>60</sup> The Australian court found that the choice of words here went “beyond the functional elements of the ideas they

56. 101 U.S. 99 (1879).

57. (1992) 173 C.L.R. 330. We discuss this case in subsection 5.1.1.2.

58. 49 F.3d 807 (1st Cir. 1995), *aff’d by an equally divided Court*, 516 U.S. 233 (1996). We discuss this case in detail in subsection 2.1.1.

59. 740 F. Supp. 37 (D. Mass. 1990).

60. See *Powerflex* at 12–13.

express, and beyond the obvious,” and that the words were “elements of expression, original and substantial, and therefore copyrightable.”<sup>61</sup> Thus, the Australian court rejected *Borland's* notion that elements of a computer program so “essential to operating something” should be outside the scope of copyright protection.<sup>62</sup>

#### *Elements Dictated by Compatibility*

In its findings regarding Bennett’s copying of the compression tables, the Australian court demonstrated its position on whether compatibility concerns could affect the protectability of program elements even more starkly. Bennett once again presented a merger defense, arguing that in order for a person using the DataFlex application development system to use a PFX-plus program, the PFXplus compression table must be identical to that of DataFlex. He further contended that this meant there was but one manner in which to express the idea of “function through compatibility,” and therefore the idea and the expression merge, making the compression table unprotectable under copyright.<sup>63</sup>

The court rejected this argument, referring once again to Judge Keeton’s *Paperback* opinion. The Australian court held that compatibility concerns could not negate the protectability of a program element:

The function of compression by means of the Hoffman method may be, and has been, performed by any one of very many different expressions in integer code. The expression given in the DataFlex table is but one of the many possible expressions. The conclusion is established that the DataFlex table is copyright [*sic*] before consideration is given to the PFXplus table. *The desire of Dr. Bennett for the compatibility he achieved by reproduction of the DataFlex table, not any inseparability of function and expression of the Hoffman compression method, constrained him to merge function and expression.*<sup>64</sup>

Viewed in this light, the Australian decision was very near the opposite side of the spectrum—in terms of protection offered to interface specifications—from judicial decisions in the United States. Perhaps the contrast can be discerned most clearly by comparing the *Powerflex* decision to the Eleventh Circuit’s ruling in *Bateman v. Mnemonics, Inc.*<sup>65</sup> In *Bateman*, the Eleventh Circuit found that, although interface specifications were not *per se* uncopyrightable, compatibility concerns would typically negate the protectability of certain elements of computer programs.

61. *Id.* at 13.

62. *Borland*, 49 F.3d at 816.

63. *Powerflex* at 17.

64. *Id.* (emphasis added).

65. 79 F.3d 1532 (11th Cir. 1996). We discuss this case in subsection 2.1.2.1.



By distinguishing constraints dictated by compatibility from constraints flowing from the inseparability of function and expression, the Australian court basically decided on a *per se* basis that no elements of computer programs dictated by compatibility concerns could be determined to be unprotected. In light of the state of U.S. law as represented by *Borland* and *Bateman*, the Australian court's reliance on the opinion of Judge Keeton in *Paperback* as an example of U.S. legal authority seems puzzling. Perhaps more mystifying is the Australian court's failure to consider an even more obvious U.S. case relevant to the questions before the court: *Computer Associates International, Inc. v. Altai, Inc.*,<sup>66</sup> which established the abstraction-filtration-comparison test for questions of infringement by non-literal copying of computer programs.

Interestingly, the Australian court made a completely contrary finding with respect to another portion of the facts presented, thereby casting doubt on the above-described holding. Bennett also copied a substantial portion of the error text table contained in the DataFlex programs. When an error occurred during use of the DataFlex software, the program referred to an error table that contained numbered errors corresponding to specific lines of text to be displayed on the computer screen, describing the error to the user. Bennett's error text lines were substantially similar to those provided in the DataFlex table. Nonetheless, in this circumstance, the court found that the "expression of the idea is inseparable from its function and is not copyrightable."<sup>67</sup> The court provided no explanation as to why it so easily applied the merger doctrine to the error text lines, but not to the words of the DataFlex language. However mixed the message of the court may be, it certainly cannot be interpreted as favorable to developers of interoperable software.

Although the court arguably misused U.S. authorities, it did not refer to any of the four recently decided software copyright cases in other Commonwealth countries: two in Canada and two in the United Kingdom.<sup>68</sup>

### 5.1.3 Australia's Software Reverse Engineering Amendment

In the spring of 1999, four years after the CLRC issued its final report, a set of copyright amendments relating to computer programs, including a

66. 982 F.2d 693 (2d Cir. 1992).

67. *Powerflex* at 15.

68. See Jonathan Band, "Computer Associates Crosses the Atlantic and Lake Ontario: *Richardson v. Flanders* and *Delrina v. Triolet*," *International Computer Law*, June 1993, at 2; Band, "*Matrox Electronic Systems v. Gaudreau*," 6 *European Intellectual Property Review* D-138 (1994).

reverse-engineering exception similar to the that of the EU Software Directive, were introduced in the Australian Parliament.<sup>69</sup> The amendments were passed by the Senate on June 29 and by the House of Representatives on August 12. They went into effect on September 30.

### 5.1.3.1 The Need for a Reverse-Engineering Exception

On August 11, 1999, in the Second Reading Speech, Attorney-General Daryl Williams explained the government's rationale for introducing the legislation. He described the growing importance of computers and computer networks to the economy. With the advent of the Internet, he said, "there is an obvious need for computers and the programs which drive them to communicate, connect, or 'interoperate' with each other."<sup>70</sup> He then explained the need for interface information in order to achieve interoperability, and how, as a technical matter, this information often can be obtained only through reverse engineering. He singled out the reverse-engineering technique known as "decompilation," which involves translating the machine-readable object code into a higher-level, human-readable format.<sup>71</sup>

Williams 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."<sup>72</sup> He 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.<sup>73</sup>

69. Copyright Amendment (Computer Programs) Act, 1999.

70. Speech on Copyright Amendment (Computer Programs) Bill 1999, Second Reading at 2.

71. *Id.* at 2–3.

72. *Id.* at 3.

73. *Id.* at 3–4.

At this point, Williams explained the provisions of the amendment: “[A]s 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.”<sup>74</sup> He hastened to add that the amendment would “not weaken the existing proscription of software piracy,” explaining that pirates do not reverse engineer but rather engage in wholesale copying.<sup>75</sup>

Finally, Williams 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.<sup>76</sup>

#### 5.1.3.2 The Structure of the Reverse-Engineering Exception

In an explanatory memorandum that accompanied the amendment, the government discussed the four alternatives it had considered.<sup>77</sup>

The first alternative was to leave the law unchanged. This was rejected for the reasons outlined in the Speech on Second Reading: the costs to the Australian software industry would be too great.

The second 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 the U.S. government and the Business Software Alliance supported this option, the Australian government did not pursue it, because of the uncertain extent of protection against infringement actions such a provision would provide to software developers. Since fair dealing, like fair use, is determined case by case by courts, the contours of the new fair-dealing provision would emerge only from lengthy and expensive litigation.<sup>78</sup>

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, insofar as a statutory exception provided more certainty to developers of interoperable software. At the same

74. *Id.* at 4.

75. *Id.*

76. *Id.* at 5–6.

77. Explanatory Memorandum on Copyright Amendment (Computer Programs) Bill of 1999 at § 4.3.

78. *Id.*

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.<sup>79</sup> 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 EU Software Directive and adding provisions concerning security testing and software-hardware interoperability.<sup>80</sup>

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

#### *Black-Box Reverse Engineering*

Section 47B(3) parallels article 5(3) of the EU 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 running a copy of the program for the purpose of studying the ideas behind the program and the way in which it functions.”

#### *Decompilation for Interoperability*

Section 47D parallels article 6 of the EU 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 that 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

79. A manufacturer of a peripheral device such as a disk drive or a printer may have to reverse engineer a computer’s operating system to ensure that the peripheral device functions properly with the computer. On why the EU Software Directive permits decompilation for software-to-hardware interoperability, see Band and Katoh, *Interfaces on Trial* at 248–249.

80. Explanatory Memorandum on Copyright Amendment (Computer Programs) Bill of 1999 at § 4.3.

that attach to the original program and products that compete with the original program.<sup>81</sup>

The other provisions of section 47D place limits on decompilation. Under subsection (c), the adaptation can be “made only to the extent reasonably necessary to obtain” the interface information. Under subsection (e), decompilation 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 EU Software Directive contains similar limitations.

Section 47D contains a significant provision not found in the EU Software Directive. Subsection (d) permits decompilation under the following condition: “to the extent that the new program reproduces or adapts the original 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 it unambiguous that a developer of interoperable software can include in the new program the interface information derived from the original program. Although this concept is implicit in the EU Software Directive (what would be the point of permitting decompilation 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.”<sup>82</sup> Section 47D(d) eliminates any ambiguity by directly permitting the copying of any element necessary for interoperability.

81. As was discussed in chapter 2, the question of attaching versus competing has long been central in the interoperability 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 distinction between attaching and competing 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 backward and forward compatibility—to ensure that the competitive product can interoperate with products on the market as well as those not yet introduced—the competition often must examine both sides of the interface.

82. We discuss this issue in greater detail in subsection 1.3.2. Under U.S. law, the unprotectability of interface information is primarily based on judicial interpretation of 17 U.S.C. 102(b). See *Computer Associates v. Altai*, 982 F.2d 693 (2d Cir. 1992); Band and Katoh, *Interfaces on Trial* at 83–165; subsection 1.3.1 and section 2.1 above.

### *Error Correction*

Section 47E permits reproducing or adapting computer programs to correct an error in the program that prevents it from operating as its author intended 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 EU Software Directive does not contain a detailed provision dealing exclusively with error correction. However, article 5(1) 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 amendment, therefore, supplies additional specificity to a concept appearing in the EU 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.

### *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 EU Software Directive does not contain a parallel provision, but the dangers posed by hacking and viruses were better understood in 1999 than in 1991 (when the EU adopted the Software Directive). Moreover, the 1998 U.S. Digital Millennium Copyright Act exempted computer system security testing from its ban on circumvention and circumvention devices.<sup>83</sup>

### *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., 47B(3), 47D, 47E, and 47F)

83. See 17 U.S.C. § 1201(j), discussed above in subsection 3.1.6.3.

“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-competition and pro-interoperability objective of the legislation. The EU Software Directive contains a similar provision in article 9(1).

In enacting a software reverse-engineering amendment, Australia chose to follow the more certain civil-code approach of the EU Software Directive. Because Australia’s software industries depend on interoperability, and interoperability often can be achieved only through reverse engineering, the government was impelled to eliminate the legal barriers to software reverse engineering.

#### 5.1.4 Australia’s Implementation of the Australia-U.S. Free Trade Agreement

In 2004, Australia and the United States entered into a free-trade agreement that required the parties to “provide adequate legal protection and effective legal remedies against the circumvention of effective technological measures that authors, performers, and producers of phonograms use in connection with the exercise of their rights and that restrict unauthorised acts in respect of their works, performances, and phonograms.”<sup>84</sup> The agreement specified that the parties shall provide legal remedies against any person who “knowingly, or having reasonable grounds to know, circumvents without authority any effective technological measure that controls access to a protected work, performance, or phonogram, or other subject matter.”<sup>85</sup> The agreement further required the parties to provide remedies against any person who “manufactures, imports, distributes, offers to the public, provides, or otherwise traffics in devices, products, or components, or offers to the public, or provides services...” that circumvent an effective technological measure. The agreement allowed the parties to adopt exceptions to these prohibitions, including one for “non-infringing reverse engineering activities with regard to a lawfully obtained copy of a computer program, carried out in good faith with respect to particular elements of that computer program that have not been readily available to the person engaged in those activities, for the sole purpose of achieving interoperability of an independently created computer program with other programs.”<sup>86</sup>

84. Australia-U.S. Free Trade Agreement, Art. 17.4.7(a), May 18, 2004.

85. *Id.* at (a)(i).

86. *Id.* at (e)(i).

In 2006, the Australian Parliament amended the Copyright Act of 1968 to implement the changes required by the Australia-U.S. FTA. The amendments included civil and criminal prohibitions on the circumvention of an access-control technological protection measure, on the manufacture or distribution of a device that circumvents a technological protection measure, or on providing a circumvention service. The amendments also provided interoperability exceptions to these prohibitions. Under these exceptions, the prohibitions on circumvention and circumvention devices do not apply when the circumvention:

- (i) relates to a copy of a computer program (the original program) that is not an infringing copy and that was lawfully obtained; and
- (ii) will not infringe the copyright in the original program; and
- (iia) relates to elements of the original program that will not be readily available to the person when the circumvention occurs; and
- (iii) will be done for the sole purpose of achieving interoperability of an independently created computer program with the original program or any other program.<sup>87</sup>

Although the FTA refers to “non-infringing reverse engineering activities . . . carried out . . . for the sole purpose of achieving interoperability,” these exceptions are not limited to reverse engineering. Rather, they apply to any circumvention activity performed to achieve interoperability between computer programs, including the circumvention of an authentication handshake designed to prevent interoperability with software products developed by other vendors. Thus, the exception in the Australian copyright law has the same breadth as the interoperability exception in section 1201(f) of the Digital Millennium Copyright Act.

## 5.2 Singapore

Singapore amended its copyright laws to permit software reverse engineering in 1998—the year before Australia adopted its own reverse-engineering

87. Copyright (Amendment) Act, 2006 §§ 116AN(3), 116AO(3), 116AP(3), 132APC(3), 132APD(3), and 132APE(3). This exception was anticipated by the CLRC Report’s discussion of the modification of “locked” programs. Although the CLRC Report recommended that the “modification of a locked program for the purpose of circumventing the lock should be prohibited” (10.94 at 175), it suggested that circumvention should be permitted for “back-up copying, interoperability, and error correction.”



amendment.<sup>88</sup> The two countries employed different legislative processes and statutory approaches, yet ended up in similar places for similar reasons. In each instance, the government made clear that the amendments were necessary to allow the domestic software industry to compete in the global market. As was discussed above, Australians pursued a lengthy and often contentious deliberative process to arrive at an amendment modeled on the EU Software Directive. In contrast, Singapore quietly developed an approach that closely follows the United States' reliance on the fair-use doctrine. Singapore's statutory amendment was necessitated by the 1996 ruling of the Singapore Court of Appeal's decision in *Aztech v. Creative Technologies*—a decision that had the affect of prohibiting software companies from engaging in reverse engineering in Singapore.

### 5.2.1 Singapore Courts and Interoperability: *Aztech v. Creative Technology* (1997)

In 1995, the High Court in Singapore relied on the U.S. decisions concerning reverse engineering to find the practice permissible under Singapore's copyright law. The following year, the Court of Appeal reversed this decision.

#### 5.2.1.1 Facts

Creative Technology had developed "Sound Blaster" sound cards for use with personal computers. Along with its sound cards, the company packaged some computer software, including a specific program known as TEST-SBC. Somewhat later, Aztech began developing a sound card that would interoperate with applications designed for use with either the Sound Blaster or other standard sound cards in the industry. As part of the process of developing its own sound cards, Aztech copied Creative Technology's TEST-SBC program into the random-access memory of its own computers. This was done to allow Aztech's research-and-development team to run the computer program and to test it by running it along with other programs. Through this process, Aztech's researchers hoped to study the manner in which the Creative Technology program communicated directions to the Sound Blaster sound card. Thus, the reverse engineering involved

88. This section is based on passages in the following publications: Jonathan Band and Taro Ishiki, "Interoperability in the Pacific Rim: Reversal of Fortunes in Singapore and Australia," *Journal of Proprietary Rights*, July 1997, at 2; Jonathan Band, "Software Reverse Engineering Amendments in Singapore and Australia," *Journal of Internet Law*, January 2000, at 17.

only running Creative Technology's program in tests (i.e., black-box reverse engineering), not the decompilation of the program from object code into a higher-level form. The legal issue presented was whether Aztech's copying of the computer program into memory, in order to run the program as part of its hardware development process, qualified as "fair dealing"—the Commonwealth equivalent to the U.S. fair-use doctrine. The court evidently assumed that a transitory RAM copy was a copy within the meaning of Singapore's copyright law.

### 5.2.1.2 The High Court's Ruling

#### *Fair Dealing*

The four enumerated factors applied in determining whether a use should be considered "fair dealing" under the Singapore Copyright Act were remarkably similar to those under section 107 of the U.S. Copyright Act. However, Singapore's statute was more restrictive in that it also required that the use of a literary work be "for the purpose of research or private study."<sup>89</sup> The statute also explicitly excluded from its definition of "research" industrial research or research carried out by companies or other business groups. But even under this restrictive definition of fair dealing the Singapore High Court found the use by Aztech to constitute "private study" within the intended meaning of the statute. "It seems to me," the judge stated, "that a study is *private* if the study and the information and knowledge acquired through it are kept or removed from public knowledge or observation and this is so even if the purpose may be of a commercial nature."<sup>90</sup> This aspect of the decision is important, since a failure to construe "private study" to reach commercially motivated study would necessarily have excluded all forms of reverse analysis of computer software in a commercial context from the fair-dealing defense under Singapore law. It is also notable in that it shows the court's stretching of the traditionally restrictive doctrine to accommodate the analysis practiced by Aztech.

In its evaluation of the four enumerated fair-dealing factors, the court emphasized that Aztech copied the Creative Technology program into the memory of its own computers only in order to run the program. The judge wrote that the program "was copied to the memory of the computer." He continued: "That is the ordinary way of running the program and it was undoubtedly the way Creative intended it to be run. It was not copied in

89. *Aztech Systems Pte Ltd v. Creative Technology Ltd* (1996) 1 SLR 683.

90. *Id.*

any other way nor in any other form. It was not disassembled or printed out. No copies of it in any form were made for distribution or which could be distributed. I think this weighs in favour of fairness.”<sup>91</sup> This was important to the first fair-dealing factor—the purpose and character of the dealing—because it indicated that the computer program was copied for the intended purpose of running the program. It was also important to the court’s determinations on the second and third factors (the nature of the work and the amount and substantiality of the part copied), since the nature of a computer program requires that it be copied in its entirety into the computer’s memory in order for the program to be run efficiently.

A significant distinction emphasized by the court several times during its fair-dealing determination was that Aztech was attempting to develop a competing sound card, not a software program that would compete with Creative Technology’s TEST-SBC program: “I think it has to be borne in mind though that the product that was developed and marketed by Aztech was the *sound card*. It was not a *software program* that emulated TEST-SBC or the instructions in it.”<sup>92</sup> The court found this to be important in evaluating not only the commercial purpose of the dealing but also the effect upon the potential market or value of the work, since the development of a competing hardware product would not compete directly, or perhaps at all, with the analyzed software.

Finally, the court also considered, as an independent and additional factor in its fair-dealing determination, the public interests advanced by the dealing at issue. Like the Ninth Circuit in *Sega*, the Singapore High Court found that increasing the competitors in the market “to more than just Creative and those licensed by it” would be a benefit “in consonance with the purpose of the Act.”<sup>93</sup> The court then concluded that the balance of the considerations required a finding that Aztech’s use of Creative Technology’s program constituted fair dealing under the act and therefore succeeded as a defense to infringement.

#### *Two Important Caveats*

The limited extent of the reverse engineering reviewed in this case must be noted. All the High Court considered was Aztech’s copying of the TEST-SBC program into RAM in order to study a specific command of the program. The court specifically observed that Aztech did not decompile Creative

91. Id.

92. Id.

93. Id.

Technology's software. This observation creates the implication that the High Court might have ruled differently in a case involving decompilation of a computer program's object code.

A second potential limitation suggested by the High Court's opinion is that reverse engineering of computer programs for the purpose of developing competing *software* might not be construed as fair dealing. Here, Aztech studied the Creative Technology program in order to develop competing *hardware*. As was stated above, this was a crucial factual distinction in the court's evaluation of two of the four statutory fair-dealing factors. The presence of direct competition might have altered the court's ultimate finding of fair dealing.

#### *Use of Foreign Law*

It is interesting that the High Court considered U.S. copyright decisions regarding reverse engineering specifically, and fair use generally, and yet left open the possibility of the limitations discussed above. The court first justified its examination of U.S. decisions because the provisions for fair dealing in Singapore and for fair use in the United States "are in many respects similar."<sup>94</sup> The U.S. decision discussed most extensively by the court was the Ninth Circuit's *Sega* decision. However, *Sega* contains neither of the limitations implicitly imposed by the High Court—the exclusion of decompilation and reverse engineering for the purpose of direct competition. The Singapore High Court instead turned to the *Sega* decision rather warily, embracing only its recognition of the public interest as relevant in fair-use determinations:

While I am conscious of the need to approach the American authorities with caution in view of the development in that jurisdiction of the distinction between *ideas and functional concepts* which are not protected by copyright and the *expression* of those unprotected elements which is protected it is at least comforting to know that the broader public interest is a factor to be taken into consideration albeit in respect of the "purpose and character" of the use although I would prefer to treat it as a separate matter to which regard is to be had.<sup>95</sup>

The court also investigated the applicability of an 1871 patent case, *Betts v. Wilmont*, in which a British court held that a purchaser of goods patented by another party has control of the goods such that, absent a clear and explicit agreement to the contrary, the purchaser may sell or use the articles however he pleases. Aztech argued that under *Betts v. Wilmont* its purchase

94. Id.

95. Id.

of the software from Creative Technology gave it the “right to use it for a *reasonable* purpose.”<sup>96</sup> The court agreed:

When a man buys a Sound Blaster sound card and with it comes TEST-SBC he expects to have and to exercise his rights of ownership over it. He can use it. That is one of the rights of ownership. He can run the software in his PC. That is what it is for. To run it the program has to be copied to the PC’s memory. He can run the program in as many PCs and as often as he pleases. He can study it to see what it is doing and he can experiment with it. That is exercising his right as an owner to use it. But he cannot make and distribute copies of it. That is not using it. That is not using the software he has bought.<sup>97</sup>

Most importantly, the court located the above-described right in Aztech’s ownership of the software, not in the reasonableness of the use for which the software was employed.<sup>98</sup> Thus, even though the court suggested potential limitations on the fair-dealing defense, it provided a fairly broad and independent defense based on the right of ownership as recognized in *Betts v. Wilmont*.

### 5.2.1.3 Reversal by the Court of Appeal

In 1997, the Court of Appeal reversed the High 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.<sup>99</sup>

#### *Fair Dealing Defense*

Section 35(1) of the act 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.” As was noted above, the High Court had ruled that Aztech’s reverse engineering was private study and not research, and therefore was permitted under section 35(1). The Court of Appeal disagreed with the High Court and decided that section 35(1) excludes commercial research as well as private study for commercial purposes. The Court of Appeal said that in order to come within the “private study” exception, the copying must be undertaken by the student. The Court of Appeal was influenced by the

96. *Id.*

97. *Id.*

98. See *id.*

99. *Creative Technology, Ltd v. Aztech Systems Pte Ltd* (1997) 1 SLR 621.

argument that if it were to adopt a broader construction of “private study” to extend to “private study for commercial purposes” it effectively would render meaningless the specific exclusion of commercial research under section 35(5). Since the Court of Appeal concluded that Aztech’s admitted copying of TEST-SBC did not qualify as “research or private study,” the fair-dealing defense was not available to Aztech.

#### *“Essential Step” Defense*

Section 39(3) of SCA, which is derived from section 117 of the U.S. Copyright Act, allows the owner of a computer program to copy or adapt that program as an essential step in the utilization of the program in conjunction with a machine. The Court of Appeal considered whether the essential steps in using a computer program could include copying it into RAM for the purpose of studying the underlying ideas and concepts of the program. Relying for guidance on interpretations of section 117 of the U.S. Copyright Act, including *Apple Computer, Inc. v. Formula International, Inc.*<sup>100</sup> and *Allen-Myland, Inc. v. IBM Corp.*,<sup>101</sup> the Court of Appeal agreed with Creative Technology that section 39(3) was enacted for the limited purpose of allowing the rightful owner of the program to load and use it in his computer. In the Court of Appeal’s view, section 39(3) did not allow copying or adaptation for the purpose of creating of a compatible product. Therefore, Aztech’s RAM copy of TEST-SBC could not be deemed an essential step in the utilization of the program. In reaching this conclusion, the Court of Appeal explicitly rejected the Fifth Circuit’s decision in *Vault Corp. v. Quaid Software Ltd.*,<sup>102</sup> which held that section 117 permitted RAM copying during the course of reverse engineering.

#### *Implied License*

Aztech argued that when it purchased TEST-SBC, it obtained along with its physical ownership the right to use it for a reasonable purpose. Aztech again relied upon the 1871 British patent case, *Betts v. Wilmont*, which held that the purchaser of a patented article has an implied license to sell the article and to use it for any reasonable purpose, absent some clear and explicit agreement to the contrary. Aztech asserted that its use was for a reasonable purpose; the copy of TEST-SBC was made to ascertain functionality

100. 594 F. Supp. 617 (C.D. Cal. 1984), *aff’d*, 725 F.2d 521 (9th Cir. 1984).

101. 746 F. Supp. 520 (E.D. Pa. 1990), *vacated and remanded*, 33 F.3d 194 (3d Cir. 1994), *cert. denied*, 513 U.S. 1066 (1994).

102. 847 F.2d 255 (5th Cir. 1988).

with the object of building a non-infringing compatible product. The trial judge made no finding on whether such use was indeed reasonable, but was persuaded that Aztech merely exercised an inherent right of ownership conferred by the purchase of TEST-SBC.

The Court of Appeal disagreed with the trial judge and held that the proposition in *Betts* is inapplicable in the Singaporean copyright context. The Court of Appeal noted that the exclusive rights granted to the patent owner differ materially from those accorded to the copyright owner and that to uphold such an implied license would run contrary to the provisions of the SCA.

#### *Decompilation of Firmware*

At trial and on appeal, Creative Technology alleged that Aztech had decompiled a substantial portion of the firmware embedded in the Sound Blaster microprocessor. The High Court decided as a factual matter that Aztech had not decompiled the firmware.

The Court of Appeal disagreed. After an extensive review of the facts, the Court of Appeal determined that Aztech had the means, the motive, and the opportunity to decompile Creative Technology's firmware. Furthermore, the literal similarities between Aztech's and Creative Technology's firmware raised the "irresistible inference that the chances of independent development on the part of Aztech were low."<sup>103</sup> However, because no more than 4 percent of Aztech's code was identical to Creative Technology's, the Court of Appeal held that Aztech's copying did not amount to a substantial taking.

Although the appellate court found no liability for the decompilation, it nonetheless seemed disturbed by the act of decompilation itself:

This . . . in no way prejudices our finding of [decompilation], which involves a degree of reproduction and adaptation having a greater impact in terms of revealing the ideas and interfaces of a copyright holder's program, insights which would not otherwise have been obtained by independent development or empirical observation within a given time frame.<sup>104</sup>

In other words, the Court of Appeal viewed the decompilation negatively precisely because it revealed features not protected by copyright.

There is a clear contrast between the Court of Appeal's hostile view toward decompilation and the Ninth Circuit's decision in *Sega Enters, Ltd.*

103. *Creative Technology, Ltd v. Aztech Systems Pte Ltd* (1997) 1 SLR 621.

104. *Id.*

*v. Accolade, Inc.*<sup>105</sup> In *Sega*, the court decided that decompilation in order to gain access to the unprotectable elements of the program, when no other means of access was available, was a fair use. The *Sega* court did not view decompilation as an independent wrong, but as a legitimate means of studying the ideas of the computer program in order to create a non-infringing product. (Interestingly, the Court of Appeal followed *Sega's* holding that section 117 did not permit decompilation.)

After the *Aztech* decision, no commercially motivated research or study was entitled to the fair-dealing defense in Singapore. Thus, developers of interoperable software could not use the fair-dealing defense to excuse the interim copies made during decompilation and other forms of reverse engineering, nor could they rely on “essential step” or “implied license” arguments to justify their interim copying.

### 5.2.2 Responding to *Aztech*: Singapore’s Software Reverse-Engineering Amendments

In response to the Court of Appeal’s decision in *Aztech*, Singapore’s Attorney-General of Law drafted an amendment to the Copyright Act, which was introduced in the Singapore Parliament in February of 1998.<sup>106</sup> 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: “[T]he 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 do not bar the use of copyright materials for commercial research.”<sup>107</sup>

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.

105. 977 F.2d 1510 (9th Cir. 1992).

106. Copyright (Amendment) Bill of 1998.

107. Second Reading of Copyright (Amendment) Bill of 1998 (February 19, 1998). On the interoperability debate in Hong Kong, see section 5.3 of the present chapter.



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.<sup>108</sup>

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.<sup>109</sup> The government sought to allow Singapore companies to develop interoperable software products.

In 2004, Singapore amended its copyright law to include provisions modeled on the black-box reverse-engineering provision and the decompilation provision of the EU Software Directive. At the same time, it left its amended fair-dealing provision in place. Furthermore, the new provisions (39A and B) specifically stated that they were “without prejudice to the generality of section 35” and “d[id] not limit the operation of that section.”<sup>110</sup> Thus, reverse-engineering activities that did not fall directly within the two new provisions could still be permitted by the fair-dealing section.

The 2004 amendments also implemented the Singapore-U.S. Free Trade Agreement, signed by the parties the previous year. Like the Australia-U.S. FTA, the Singapore-U.S. FTA required the parties to provide effective legal remedies against the circumvention of technological protection measures.<sup>111</sup> Similarly, the Singapore-U.S. FTA permitted parties to provide a reverse-engineering exception to the circumvention prohibition. Thus, under section 261D(1)(d) of the Singapore Copyright Act, the prohibitions

108. Second Reading of Copyright (Amendment) Bill of 1998 (February 19, 1998).

109. As was discussed above in chapters 1 and 2, the U.S. Court of Appeals for the Ninth Circuit, in *Sega Enter. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1527–1528 (9th Cir. 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.”

110. Copyright Act, Singapore Statutes Chap. 63.

111. See subsection 5.1.4 of the present chapter.

on circumvention and circumvention devices do not apply when the circumvention is done

- i. in good faith;
- ii. in relation to a copy of a computer program that is not an infringing copy; and
- iii. with respect to particular elements of the computer program that are not readily available to the person doing the act,

for the sole purpose of achieving interoperability of an independently created computer program with another computer program. . . .<sup>112</sup>

### 5.3 Hong Kong

In the months before the turnover of Hong Kong to China, the Hong Kong Legislative Council busily worked at revising its copyright laws with the understanding that the new laws would continue in effect after the departure of the British on July 1, 1997.<sup>113</sup> One of the provisions the Legislative Council considered would have specifically permitted decompilation for purposes of achieving interoperability. Rather than let the Legislative Council determine for itself what was in the best interests of Hong Kong, U.S. software companies represented by the Business Software Alliance<sup>114</sup> encouraged the U.S. Trade Representative to pressure the Legislative Council into dropping the provision. The Legislative Council ultimately abandoned the decompilation provision, but in its place adopted, with BSA's blessing, another provision based on the U.S. fair-use doctrine. This fair-use language arguably permits more reverse engineering than the decompilation provision would have permitted.

#### 5.3.1 The November 1996 Copyright Bill Consulting Paper

In November of 1996, the Department of Intellectual Property of Hong Kong's Board of Trade and Industry produced a consultation paper on a new copyright law, which included a proposed text for the bill. Section 57 of the Consultation Paper draft specifically permitted decompilation of object code for purposes of achieving interoperability. Section 57(1)(a) defined decompilation as the conversion of a "computer program expressed

112. Copyright Act, Singapore Statutes Chap. 63.

113. This section is based on 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*, February 1998, at 8.

114. The BSA's members included Microsoft, Novell, Autodesk, and Lotus (a subsidiary of IBM).

in a low level language . . . into a version expressed in a higher-level language.”<sup>115</sup> The only permitted objective of the decompilation was obtaining “the information necessary to create an independent program which can be operated with the program decompiled or with another program.” Section 57(3) prohibited decompilation when the “information necessary to achieve the permitted objective” was “readily available” to the user by other means, when the decompiling was not confined “to such acts as are necessary to achieve the permitted objective,” when the information obtained by decompilation was supplied to a person not essential to achievement of the permitted objective, or when the information obtained by decompilation was used “to create a program which is substantially similar in its expression to the program decompiled.” Finally, section 57(4) expressly voided contractual restrictions on decompilation otherwise permitted by section 57.

Section 57 of the Consultation Paper closely followed the decompilation provision of the UK copyright statute, section 50B, which closely followed the article 6 of the EU Software Directive. In other words, from a substantive perspective there was nothing controversial about section 57. Nonetheless, section 57 drew a sharp reaction from the U.S. government. In a cable sent in December of 1996, the U.S. government recommended deletion of section 57. The United States specifically objected to the voiding of contractual restrictions on decompilation, noting that this stood “in marked contrast to the EC Software Directive.” The cable explained that “a similar restriction on the freedom of contract was proposed and rejected during consideration of the directive.” In fact, the opposite was true. Article 9(1) of the directive explicitly provides that “[a]ny contractual provisions contrary to Article 6 [the decompilation provision] . . . shall be null and void.” Thus, an official communication between the U.S. government and the Hong Kong Department of Intellectual Property contained an obvious legal error.

The cable suggested that section 57 had a broader permitted objective than article 6 of the EU Software Directive in that section 57 permitted decompilation for the purpose of creating new programs, whereas the EU Software Directive permits decompilation only for the purpose of achieving interoperability of existing programs. However, this interpretation of article 6 of the directive as permitting decompilation only to achieve interoperability between two existing products, and not to develop a new interoperable product, had no basis in the text of the directive or in its legislative history.

115. It is this act of conversion or translation that implicates the copyright owner’s right to create a derivative work.

The cable also indicated that a decompilation provision would be inappropriate in view of the challenges of enforcing copyrights in Hong Kong. Yet the cable failed to explain how permitting decompilation would weaken enforcement. Good enforcement flows from an effective judicial system, and decompilation, of course, would have no effect on Hong Kong's judiciary. A decompilation provision also would not complicate judicial proceedings; section 57 would not provide a colorable defense to a software "pirate" engaged in wholesale copying.

The cable concluded by suggesting that Hong Kong delete section 57 because several other countries in the Asia-Pacific region, including Australia, New Zealand, Japan, and Korea, had considered and rejected a decompilation provision based on the EU Software Directive. This, too, was a misstatement. The Australian Copyright Law Review Committee in 1995 recommended legislative enactment of a decompilation provision,<sup>116</sup> and, as was discussed above, in 1999 Australia amended its copyright law to include a decompilation exception based on the EU Software Directive. In Japan, a private-sector advisory committee began to consider a reverse-engineering exception at the request of the Cultural Affairs Agency. The CAA's simple act of making the request, however, drew a sharp rebuke from the U.S. government, and the advisory committee concluded that it was premature to consider the issue.<sup>117</sup> Similarly, as will be discussed below, the Korean Ministry of Information and Communications suspended its consideration of a reverse-engineering exception in the face of protests by the U.S. government.<sup>118</sup> In short, neither Japan nor Korea rejected a decompilation provision after considering its merits; rather, they abandoned their efforts in response to U.S. pressure, but left the door open to revisiting the issue in the future.<sup>119</sup>

Although the U.S. government urged the Department of Intellectual Property to discard section 57, the American Committee for Interoperable Systems<sup>120</sup> pressed for amendments to make section 57 even more conducive to interoperability. Specifically, ACIS noted (in comments dated November 27, 1996) that section 57 contained an ambiguity that was

116. See subsection 5.1.1 of the present chapter.

117. See Band and Katoh, *Interfaces on Trial* at 297–316.

118. See section 5.5 of the present chapter.

119. Additionally, most scholars agree that software reverse engineering is currently permitted under Japanese law. See, e.g., Ozaki, "Copyright Protection of Software: The Japanese View," 1990 *Computer Law Reporter* 950 (1990).

120. The members of ACIS included Sun Microsystems, Storage Technology Corporation, 3Com, Fujitsu Systems Business of America, and NCR.

present in article 6 of the EU Software Directive. Both provisions permitted decompilation to achieve interoperability between two software products, but were silent about decompilation to permit interoperability between software and hardware. ACIS observed that the French National Assembly rectified this problem when it implemented the EU Software Directive by explicitly permitting decompilation to achieve software-hardware interoperability. Likewise, the Australian CRLC's decompilation provision applied to software-hardware interoperability. ACIS also urged that section 57 be extended to permit decompilation for purposes of error correction, again citing the Australian CLRC as a precedent.

### 5.3.2 The First Reading in the Legislative Council

In late February of 1997, the Board of Trade and Industry submitted its Copyright Bill to the Legislative Council, and it had its first reading. The bill retained the decompilation provision, which now appeared as section 60. The new section 60 differed from the original section 57 in one significant respect; whereas original section 57(4) voided contractual restrictions on decompilation, new section 60 expressly stated that the decompilation privilege was "subject to any agreement to the contrary." In other words, the bill on first reading took the completely opposite position on this issue from the Consultation Paper.

On April 11, 1997, ACIS submitted comments to the Legislative Council's Bills Committee applauding the retention of a decompilation exception. ACIS noted the consistency of section 60 with the EU Software Directive, with the case law in the United States treating decompilation as a fair use, and with the Australian CLRC's recommendations.

ACIS proceeded to observe that section 60 differed from its antecedents on the issue of the enforceability of contractual restrictions on reverse engineering. After quoting article 9(1) of the EU Software Directive, which voided contractual restrictions on decompilation, ACIS explained that "[t]he European Community included article 9(1) in the Directive because it correctly understood that without such a provision, software companies with market power would undo the delicate balance reached in the Directive by routinely including in their licenses clauses overriding Article 6." ACIS concluded that "Section 60(4) as a practical matter renders the rest of Section 60 a nullity."

### 5.3.3 The Bills Committee Hearings

Had the Business Software Alliance simply endorsed section 60 as introduced by the Board of Trade and Industry, it probably would have won

the day. Whereas section 60 theoretically permitted decompilation for purposes of achieving interoperability, section 60(4) allowed members of BSA to prohibit decompilation by shrinkwrap license.

BSA, however, continued to lobby for complete elimination of the decompilation privilege. By pitting itself against the Board of Trade and Industry, it set in motion a process that resulted in a legal framework far more liberal than section 60 on first reading, and indeed arguably more liberal than section 57 of the Consultation Paper.

During April and May of 1997, the Bills Committee held a series of public hearings on the Copyright Bill. The Business Software Alliance and the Software Publishers Association testified against section 60; ACIS testified in its favor, provided that section 60(4) was deleted or reversed. (Emery Simon, a former official in the Office of the U.S. Trade Representative, testified for BSA; Peter M. C. Choy, Deputy General Counsel of Sun Microsystems, testified on behalf of ACIS. Transcripts of the hearings are not available.) On April 18, BSA filed a detailed written submission, to which ACIS replied point by point on April 25. BSA made the following arguments:

1. Decompilation reveals the source code of a computer program.
2. Decompilation is cheaper than obtaining a license and easier than developing a new program from the beginning.
3. Decompilation facilitates “hidden” piracy by allowing the reverse engineer to copy a program’s functionality.
4. Decompilation is not necessary for interoperability because copyright owners make available the information necessary for interoperability.
5. Decompilation is not necessary because there are non-infringing means of obtaining interoperability information.
6. In the five years since the European Union adopted the Software Directive, no other country has followed its lead on decompilation.
7. A decompilation provision will suggest to the international community that Hong Kong is not serious about copyright infringement.<sup>121</sup>

ACIS responded as follows:

1. Decompilation at most reveals a shadow of the original source code, because it cannot recover the programmer’s comments nor restore the original sequence of the code. Additionally, decompilation does not yield

121. Presentation of the Business Software Alliance to the Bills Committee (April 18, 1997). As we discussed in subsection 2.2.3, Judge Jackson disagreed with many of these contentions in his decision in *U.S. v. Microsoft*, 84 F.Supp.2d 9 (D.D.C. 1999).

instructions in a high level programming language, but only in assembly language.

2. While the actual act of decompilation is cheap and easy, the engineer using decompilation must still invest significant resources to understand the jumbled, decompiled code.

3. Decompilation does not facilitate hidden piracy because copying a program's functionality, but not its code, does not constitute copyright infringement.

4. While some software developers may be willing to license interface information if doing so is consistent with their business plan, there often are circumstances when a firm may not be willing to license the information on reasonable terms. Moreover, even when a firm does license the information on reasonable terms, the information may be incomplete or untimely.

5. Sometimes other reverse engineering techniques—so called black-box reverse engineering—will reveal the interface information necessary for interoperability, but sometimes decompilation is the only effective method.

6. In addition to the then fifteen members of the EU, eight European countries had adopted a decompilation exception based on article 6 of the Software Directive—Norway, Switzerland, Bulgaria, Estonia, Poland, Romania, Russia, and Slovenia. Further, courts in four different federal circuits in the U.S. had ruled that decompilation constitutes a fair use.<sup>122</sup>

7. Given the lawfulness of decompilation in twenty-four nations, ranging from highly industrialized countries such as the U.S., the U.K., Germany and France, to much less developed countries such as Bulgaria and Romania, the world community would not view Hong Kong's adoption of a decompilation provision as a signal that it did not take copyright seriously. In this context, ACIS noted that BSA's European counsel had described article 6 of the Directive as a "reasonable exception," and that BSA had not objected to its adoption throughout Central and Eastern Europe, where copyright infringement also was a serious concern.<sup>123</sup>

### 5.3.4 The Imposition of a Fair-Use Approach

When BSA representative Emery Simon testified before the Legislative Council on April 18, he suggested that decompilation be handled case by

122. *Sega Enters. Ltd. v. Accolade Inc.*, 977 F.2d 1521 (9th Cir. 1992); *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832 (Fed. Cir. 1992); *Bateman v. Mnemonics Inc.*, 79 F.3d 1532 (11th Cir. 1996); *DSC Communications v. DGI Techs.*, 898 F. Supp. 1183 (N.D. Tex. 1995), *aff'd*, 81 F.3d 597 (5th Cir. 1996).

123. ACIS Response to BSA's April 18 Presentation to the Bills Committee (April 25, 1997).

case under Hong Kong's fair-dealing exception. The Legislative Council directed the Board of Trade and Industry to delete section 60 and to prepare amendments to the fair-dealing provisions that would accommodate decompilation in appropriate circumstances. The staff of the Board of Trade and Industry's Department of Intellectual Property proposed adding the following language to section 37 concerning fair dealing: "The incidental copying by a lawful user of a computer program in the course of research or private study is fair dealing if it is done for the purpose of studying the operation of the program under study, or of creating another independent program which is compatible with, but not substantially similar to or adapted from the program under study." In essence, the proposal imported the language of section 60 into the fair-dealing provision.

Exactly what happened next is unclear, but it is rumored that when BSA learned on May 14 of the fair-dealing amendment proposed by the Department of Intellectual Property, it requested that the Office of the U.S. Trade Representative intervene. A senior official in the Office of the USTR called the Secretary of Trade and Industry and insisted that the Board of Trade and Industry replace its fair-dealing amendment with language from the fair-use provision of the U.S. Copyright Act, 17 U.S.C. § 107. The staff of the Department of Intellectual Property then prepared the following language as a new section 37(3):

In determining whether any dealing with a work of any description is fair dealing, the factors to be considered include—

- a. the purpose and nature of the dealing;
- b. the nature of the work; and
- c. the amount and substantiality of the portion dealt with in relation to the work as a whole.

Additionally, the Department of Intellectual Property staff proposed a new section 36(2A): "In determining whether or not an act is permitted, the primary consideration is whether the act conflicts with a normal exploitation of the work and unreasonably prejudices the legitimate interests of the copyright owner."

ACIS and BSA agreed to this language. On June 24, when the Secretary of Trade and Industry submitted the bill—including this language—for its second reading, she issued a statement explaining that the language was intended to implement Hong Kong's policy with respect to decompilation. After describing the decompilation provision in the bill gazetted in March, she alluded to the concerns raised by ACIS and BSA: "[W]e accept that the decompilation provision as drafted would be so limited as to be of little practical help to software companies wanting to decompile. On the other



hand, as an exception to copyright restriction, the provision has aroused serious concerns amongst leading software companies.”<sup>124</sup>

The Secretary of Trade and Industry stated that the Board of Trade and Industry had reviewed its policy intention on decompilation, and had concluded that it “would like to encourage competition in the information technology industry by facilitating timely access to information and ideas underlying computer programs.” She continued:

Doing so is necessary for the independent creation of new products that attach to or compete with the programs under study. We accept that the incidental copying of a computer program by a lawful user during the course of decompilation or other reverse engineering performed to understand the operation of the program under study, or to develop a product inter-operable with the program under study, need *not* be absolutely restricted by copyright. Nor should it be completely deregulated. In determining whether the act should be allowed, we believe the overriding test is whether such act conflicts with the normal exploitation of the work by the copyright owner and unreasonably prejudices the legitimate interests of the copyright owner.<sup>125</sup>

The Secretary of Trade and Industry then explained how the modifications to the fair-dealing provisions flowed from these conclusions:

The object is to allow decompilation to be deemed a fair use provided it does not conflict with the normal exploitation of the rights and legitimate interests of the copyright owner. Drawing from the relevant provisions in the United States, we propose that other factors, including the purpose and nature of the dealing, the nature of the copyrighted work, and the amount and substantiality of the portion dealt with in relation to the copyright work as a whole, will also be taken into account in determining what constitutes “fair use.”<sup>126</sup>

The Legislative Council gave the bill its second reading on June 24, 1997, and its third reading on June 27. It took immediate effect on its third reading. Under the turnover agreement with China it will remain in effect for 50 years.

### 5.3.5 Fair Use vs. Fair Dealing vs. the EU Software Directive

Although the proposed section 60 explicitly permitted decompilation, it did so only for purposes of achieving interoperability, only if there was no other way of obtaining the necessary information, and, most significantly,

124. Speech by Secretary of Trade and Industry on Resumption of Second Reading Debate at 10 (June 24, 1997). Under Hong Kong law, such floor statements constitute legislative history upon which courts should rely when interpreting a statute.

125. *Id.*

126. *Id.* at 11.

only if it was not prohibited by a license term. By contrast, section 37(3) as enacted by the Legislative Council is far broader. The speech by the Secretary of Trade and Industry makes clear that decompilation could be lawful when conducted “to understand the operation of the program under study” as well as “to develop a product inter-operable with the program under study.” Thus, section 37(3) recognizes legitimate objectives beyond achieving interoperability. This expansive reading of section 37(3) is even more compelling when its origins are considered. A Hong Kong court applying this language to a case involving decompilation will understand that it derives from the U.S. fair-use doctrine, and presumably will import the interpretation of the doctrine given by the U.S. courts. The Ninth Circuit ruled in *Sega v. Accolade* that the fair-use doctrine permitted decompilation not only for purposes of learning the information necessary for interoperability, but for any legitimate reason: “[W]here 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.”<sup>127</sup>

When section 37(3) was enacted, its terms were more expansive than those of the fair-use doctrine codified at 17 U.S.C. § 107.<sup>128</sup> Section 37(3) omitted the phrase “including whether such use is of a commercial nature or is for nonprofit educational purposes” found in section 107(1), thereby eliminating the possibility of a presumption that commercial uses (such as decompilation by a profit-maximizing software firm) are unfair.

Section 37(3) also omitted the phrase “the effect of the use upon the potential market for or value of the copyrighted work” (found in section 107(4)) and inserted in its place language from article 9(2) of the Berne Convention: “whether the act conflicts with the normal exploitation of the work and unreasonably prejudices the legitimate interests of the copyright owner.” Both the statute and the statement of the Secretary of Trade and Industry emphasize that this is the primary, overriding test. The inclusion of this language did not signal an intent that section 37(3) had narrower application than article 6 of the EU Software Directive, for article 6(3) itself contains this same language. According to William Cornish of Cambridge University, this language is implicated only if decompilation is

127. The terms “disassembly” and “decompilation” have the same meaning in legal parlance, but “disassembly” is commonly used in U.S. court decisions, whereas “decompilation” is used outside the U.S.

128. In 2007, Hong Kong amended its fair-dealing provision to track 17 U.S.C. § 107 more closely.

used to develop a program substantially similar *in expression* to the decompiled program.<sup>129</sup>

Section 37(3) is broader than either of the earlier decompilation provisions (57 and 60) in another significant respect. The earlier provisions referred to decompilation, but were silent about the lawfulness of the copying that occurs during the course of black-box reverse engineering—for example, the making of interim copies of the program in RAM when the program is run for the purpose of observing its operation. Section 37(3), by contrast, is broad enough to excuse such incidental copying. Once again, the statement by the Secretary of Trade and Industry confirms this by referring to the incidental copying “during the course of decompilation or *other reverse engineering*.”

Finally, section 60 specifically permitted contractual limitations on the decompilation privilege. Section 37, by contrast, is silent on this issue, leaving open the possibility that a Hong Kong court could determine that enforcing such a limitation would undermine the “policy intention” of “encourag[ing] competition in the information technology industry by facilitating timely access to information and ideas underlying computer programs.”

Since the Business Software Alliance had acknowledged that article 6 of the EU Software Directive was a reasonable compromise in the European context, and since courts in four circuits had found decompilation to be a fair use, it is not clear why BSA so strongly preferred fair use to article 6 in Hong Kong. By all appearances, the BSA gained nothing when it pressured countries to follow the fair-use model rather than the EU Software Directive model.

#### 5.4 The Philippines

In 1997, the government of the Philippines proposed a reverse-engineering exception based on the EU Software Directive. Not surprisingly, that proposal encountered fierce opposition from the Office of the U.S. Trade Representative (USTR) and the Business Software Alliance. In response, the Philippines crafted a hybrid of the fair-use provision of the U.S. Copyright Act and article 6 of the EU Software Directive:

The fair use of a copyrighted work for criticism, comment, news reporting, teaching including multiple copies for classroom use, scholarship, research, and similar

129. William R. Cornish, “Computer Program Copyright and the Berne Convention,” in *A Handbook of European Software Law* (1993). The same language also appears in article 13 of the TRIPS Agreement.

purposes is not an infringement of copyright. *Decompilation, which is the reproduction of code and translation of the form of the computer program indispensable to obtain the information necessary to achieve the inter-operability of an independently created computer program with other programs may also constitute fair use.* In determining whether the use made of a work in any particular case is fair use, the factors to be considered shall include:

- a. The purpose and character of the use, including whether use is of a commercial nature or is for non-profit educational purposes;
- b. The nature of the copyrighted work;
- c. The amount and substantiality of the portion used in relation to the copyrighted work as a whole; and
- d. The effect of the use upon the potential market for value of the copyrighted work.<sup>130</sup>

As was noted above, when the Hong Kong Department of Intellectual Property had proposed a similar grafting of reverse-engineering language onto the fair-use provision, the USTR voiced strong opposition. But there is no public record of the USTR's objecting to the Philippine approach. As with the provision adopted in Hong Kong, it is far from clear that this formulation constrains reverse engineering more than article 6 of the directive; indeed, the opposite may be the case. The provision was enacted in June of 1997 and took effect in January of 1998.

## 5.5 Korea

Though one may question the wisdom and the propriety of the heavy-handedness of the USTR and the Business Software Alliance in Hong Kong and in the Philippines, those American interventions were far less successful than similar efforts in Japan and Korea. The Japanese Cultural Affairs Agency and the Korean Ministry of Information and Communications suspended consideration of a reverse exception in 1994 and 1995, respectively, in the face of U.S. opposition. The history of the Japanese deliberations of this issue is well documented<sup>131</sup> and will not be discussed further here. By contrast, little has been written about the Korean deliberations. What follows is a brief summary.

The Korean episode began in early May of 1995, when the Ministry of Information and Communication of the Republic of Korea (i.e., South

130. Intellectual Property Code of the Philippines, § 185.1 (emphasis added).

131. See Band and Katoh, *Interfaces on Trial* at 297–310. See also Crystal D. Talley, "Japan's Retreat from Reverse Engineering: An Unnecessary Surrender?" 29 *Cornell International Law Journal* 807 (1997).

Korea) proposed consideration of a reverse-engineering exception in the Computer Program Protection Act. The provision would excuse reproductions “where the program is temporarily reproduced or translated within the limits required for purposes of research and analysis necessary for the creation of interoperable programs.”

BSA promptly submitted lengthy comments opposing the amendment. Although the proposal by the Ministry of Information and Communication addressed software reverse engineering generally, BSA’s comments focused on decompilation. Decompilation, according to BSA, wasn’t necessary, because there were other ways to obtain interface information, including other reverse-engineering techniques and licensing; decompilation would facilitate disguised piracy and thereby harm the Korean software industry and prevent foreign investment in Korea; the *Sega* decision was “the view of only one U.S. court on a matter of first impression decided on an incomplete factual record and on a particular set of facts”; the level of protection currently afforded represented a further erosion; and the proposed exception was far broader than that in the EU Software Directive.<sup>132</sup>

The U.S. government also strongly protested the Ministry of Information and Communication’s proposal. A position paper prepared by the U.S. Patent and Trademark Office made three points that paralleled those of the BSA. First, “Korea’s current system of protection is too weak to permit any limitation on protection in the form of a decompilation exception.” Second, the EU Software Directive’s decompilation provision was far narrower than the proposed Korean exception. In particular, under the directive “[d]ecompilation can never be used to create a program that competes with the program which is being decompiled.” Third, the *Sega and Atari* decisions were read very narrowly, with the conclusion that “[d]ecompilation could not be used to create programs which competed with the programs being decompiled.”<sup>133</sup>

The American Committee for Interoperable Systems, in letters to the PTO and the USTR, responded sharply to what it called “errors” in the U.S. government’s position paper. ACIS first referred to the legislative history of the EU Software Directive and its implementations in the member states to

132. Business Software Alliance Position Paper Submitted to the Ministry of Information and Communications of the Republic of Korea on proposed Exceptions from Protection for Computer Programs Under the Computer Program Protection Act (May 2, 1995).

133. U.S. Government Views on Decompilation Position in Korean Computer Program Law (June 12, 1995).

demonstrate that article 6 of the directive permitted decompilation for purposes of developing both attaching *and* competing programs. Next, ACIS rebutted the U.S. government's parsimonious reading of *Sega* and "improper trivializ[ation]" of *Atari*. Finally, though ACIS explicitly declined to address programs of piracy in Korea, it directly challenged the U.S. government's suggestion that the decompilation could facilitate disguised piracy.<sup>134</sup>

In a letter to the Office of the U.S. Trade Representative, the Computer & Communications Industry Association "object[ed] in the strongest terms possible to the Government's flawed analysis of the EU Directive and the *Sega* decision." "We are," CCIA continued, "especially concerned that the Government is making representations to foreign governments that do not reflect an accurate and balanced analysis of the developing international jurisprudence relating to the permissibility of disassembly."<sup>135</sup>

As ACIS and CCIA corresponded with the U.S. government, the European Committee for Interoperable Systems and the Supporters of Interoperable Systems in Australia communicated directly with the Ministry of Information and Communication. ECIS suggested that "[a] provision permitting reverse engineering for interoperability purposes will facilitate legitimate competition and further consumer welfare without permitting piracy." Likewise, SISA applauded the Ministry of Information and Communication's simple exemption for reverse engineering to achieve interoperability.

In late July of 1995, the president of the Republic of Korea traveled to Washington to participate in the dedication of the Korean War Memorial on the National Mall. Secretary of Commerce Ron Brown used the opportunity to complain to his Korean counterpart about the Ministry of Information and Communication's reverse-engineering proposal. Soon thereafter, the MIC abandoned the proposal.

## 5.6 Israel

In November of 2007, Israel's legislature, the Knesset, enacted a new copyright law. The law included both a fair-use provision based on 17 U.S.C.

134. Letter from Peter M. C. Choy, Chairman, ACIS, to Bruce Lehman, Commissioner, U.S. Patent and Trademark Office (June 26, 1995); letter from Peter M. C. Choy, Chairman, ACIS, to Thomas Robertson, Assistant General Counsel, Office of the U.S. Trade Representative (June 26, 1995).

135. Letter from Gregory E. Gorman, Government Affairs Manager, CCIA, to Thomas Robertson, Assistant General Counsel, Office of the U.S. Trade Representative (July 5, 1995).

§107, and an exception for software reverse engineering based on the EU Software Directive. Indeed, the reverse-engineering exception in section 24 appears similar to the expanded version of the EU Software Directive that Australia had adopted. It permitted the making of derivative works (i.e., decompilation) for purposes of error correction and security testing, and also for purposes of obtaining the information necessary for interoperability:

(c) Copying of a computer program, or making a derivative work there from, is permitted for a person who possesses an authorized copy of the computer program, for the following purposes and to the extent necessary to achieve said purposes:

- (1) Use of the computer program for purposes for which it was intended, including correction of errors in the computer program or making it interoperable with a computer system or with another computer program;
- (2) Examination of the information security in the program, correction of security breaches and protection from such breaches;
- (3) Obtaining information which is needed to adapt a different and independently developed computer system or program, in such a way that it will be interoperable with the computer program.

(d) The provisions of subparagraph (c) shall not apply with respect to the copying of a computer program or the making of a derivative work there from, as stated in said subparagraph, if the information which has been obtained through the aforementioned means was used in a manner set forth below, or where such information was readily available without use of the aforesaid means:

- (1) The said information is transmitted to another person for a purpose different than the purposes set forth in subparagraph (c);
- (2) The said information is used to make a different computer program which infringes copyright in the said computer program.

Although in the 1990s the U.S. government urged countries to adopt a general fair-use provision instead of a specific reverse-engineering exception based on the EU Software Directive, it subsequently rejected the requests of U.S. Internet companies to include fair use in bilateral free-trade agreements. Search engines based in the United States rely on fair use to permit the copying they perform to assemble their search databases and to display search results.<sup>136</sup> The copyright regimes of other countries are not as hospitable to search engines; a Belgian court, for example, imposed copyright liability on Google for search activities that would be considered fair use in the United States. To secure the legal environment necessary for them to

136. See Jonathan Band, "Google and Fair Use," 3 *Journal of Business & Technology Law* 1 (2008).

increase their presence overseas, U.S. search-engine companies asked the Office of the U.S. Trade Representative to help “export” fair use to other countries. But in 2007, USTR denied the request that it include fair use in its “template” for future free-trade agreements:

The precise language that is used to achieve our objectives in this area is likely to vary from one trade agreement to another. . . . For example, our assessment of the commitment and capacity of a given trading partner to ensure effective protection and enforcement of the copyright-protected creations may be a factor in shaping our approach in future negotiations. We believe determining the best approach to these issues will continue to require a careful analysis based on each trading partner’s copyright system. . . .<sup>137</sup>

Here USTR implied that in most countries a fair-use provision would be abused.

Patrick Leahy, chairman of the Senate Judiciary Committee, had a different view. He stated that the Trade Promotion Authority Act of 2002

instructed the administration to negotiate agreements that provide strong protection for new and emerging technologies and new methods of transmitting and distributing products embodying intellectual property. This, too, is an objective I support. Under our laws, many such new technologies and consumer devices rely, at least in part, on fair use and other limitations and exceptions to the copyright laws. Our trade agreements should promote similar fair use concepts, in order not to stifle the ability of industries relying on emerging technologies to flourish.<sup>138</sup>

In December of 2009, the Obama administration articulated a position similar to Senator Leahy’s. At a session of the Standing Committee on Copyright and Related Rights of the World Intellectual Property Organization, Justin Hughes, the head of the U.S. delegation, stated:

We recognize that some in the international copyright community believe that any international consensus on substantive limitations and exceptions to copyright law would weaken international copyright law. The United States does not share that point of view. The United States is committed to both better exceptions in copyright law and better enforcement of copyright law. Indeed, as we work with countries to establish consensus on proper, basic exceptions within copyright law, we will ask countries to work with us to improve the enforcement of copyright. This is part and parcel of a balanced international system of intellectual property.<sup>139</sup>

137. Letter from Stanford K. McCoy, Acting Assistant U.S. Trade Representative, to Duane Webster, Executive Director, Association of Research Libraries (July 19, 2007).

138. Cong. Rec. S14720, December 4, 2007 (Statement of Sen. Leahy).

139. United States of America, Statement on Copyright Exceptions and Limitation for Persons with Print Disabilities, World Intellectual Property Organization, Standing Committee on Copyright and Related Rights, 19th Session (December 15, 2009) at 5.



This is a section of  
[doi:10.7551/mitpress/9780262015004.001.0001](https://doi.org/10.7551/mitpress/9780262015004.001.0001)

# Interfaces on Trial 2.0

**By: Jonathan Band, Masanobu Katoh**

## **Citation:**

*Interfaces on Trial 2.0*

**By: Jonathan Band, Masanobu Katoh**

**DOI: 10.7551/mitpress/9780262015004.001.0001**

**ISBN (electronic): 9780262295543**

**Publisher: The MIT Press**

**Published: 2011**



**The MIT Press**

© 2011 Massachusetts Institute of Technology. All rights reserved. Subject to the exception immediately following, this book may not be reproduced, in whole or in part, including the illustrations, in any form (beyond that copying permitted by Sections 107 and 108 of the U.S. Copyright Law and except by reviewers for the public press), without permission from the publishers. An online version of this work is available under a Creative Commons Attribution—Noncommercial—No Derivatives Works 3.0 Unported License. It can be accessed at <http://mitpress.mit.edu/band>.



For information about quantity discounts, email [special\\_sales@mitpress.mit.edu](mailto:special_sales@mitpress.mit.edu).

Set in Stone Sans and Stone Serif by the MIT Press. Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Band, Jonathan.

Interfaces on trial 2.0 / Jonathan Band and Masanobu Katoh.

p. cm.— (Information society series)

Includes bibliographical references and index.

ISBN 978-0-262-01500-4 (hardcover : alk. paper)

1. Copyright—Computer programs. 2. Computer software industry—Law and legislation. I. Katoh, Masanobu. II. Title.

K1443.C6B363 2011

346.04'82—dc22

2010017746

10 9 8 7 6 5 4 3 2 1