

# ANALYSIS



## The Case for Decompilation:<sup>1</sup> A Response to the Law Reform Commission's Report on Copyright

### Introduction

After more than six years of study, the Law Reform Commission (LRC) finally produced its report on 'reform of the law relating to copyright' ('LRC report') in January this year. In an earlier article, written before the report was released,<sup>2</sup> I assessed Hong Kong's copyright protection for computer software against international standards,<sup>3</sup> and pointed out six major shortcomings of the territory's software protection system.<sup>4</sup> The LRC report offers solutions in respect of only five, but none as regards the user's right to decompile and computer program for the purpose of achieving interoperability. This right is firmly rejected by the LRC. The only explanation given in the report is that '[the LRC] take the view that a limited right to decompilation is tantamount to creating another exception to infringement and [the LRC] do not recommend any such limited right.'<sup>6</sup>

The LRC's reasoning on decompilation may be termed the 'simplicity' argument as its essence is to keep the copyright system as simple as possible by minimising the number of exceptions to infringement. While the LRC's intention is understandable, one wonders if they might have overlooked the complexity and subtlety of the issue of decompilation and have dismissed it too hastily. Indeed, the LRC report contains no discussion at all on the issue<sup>7</sup> and the only reference to it is in the sentence quoted above.<sup>8</sup>

In the following, the issue of decompilation will be examined more closely within the framework of copyright law. The current status of decompilation in

<sup>1</sup> Also known as 'disassembly.' The term refers to the process of converting a computer program from machine-readable code to human-readable code. For a more detailed explanation, see below.

<sup>2</sup> K H Pun, 'Copyright Protection for Computer Software: Does Hong Kong Meet International Standards?' (1994) 24 HKLJ 56.

<sup>3</sup> As represented by three sets of laws: the 1978 WIPO Model Provisions on the Protection of Computer Software, the US Copyright Act of 1976, and the 1991 EC Council Directive on the Legal Protection of Computer Programs ('Software Directive').

<sup>4</sup> Note 2 above, p 78.

<sup>5</sup> 'Interoperability' is the ability to enable a piece of software or hardware to work with other software and hardware.

<sup>6</sup> LRC report, para 13.33.

<sup>7</sup> Nor did the Copyright Sub-committee's consultative document published in 1990 contain any discussion on the same topic.

<sup>8</sup> The LRC almost touched on the issue when they discussed a limited right to copy a computer's operating system for an application program to work with the computer: paras 13.31-2. Unfortunately they did not go a step further and consider the more general question of *intermediate* copying of any program for the purpose of achieving interoperability (the kind of copying necessary for decompilation). Had they done so, they would have had a more thorough discussion of the issue.

Hong Kong will first be discussed, and then contrasted with that in the two major trading partners of Hong Kong: the US and the European Community. The main arguments against decompilation will then be assessed in the context of Hong Kong, followed by the conclusion of the analysis. Contrary to the LRC report, this article concludes that a limited right to decompilation should be provided as an exception to copyright infringement in Hong Kong. Moreover, the right should be granted regardless of whether the final product competes with the decompiled program.

### Decompilation: meaning and present status

Decompilation of computer programs means 'reproduction and adaptation (translation) of computer programs into a form in which the coding and structure of the program can be examined and analysed.'<sup>9</sup> Typically the procedure converts a computer program in machine-readable object code to a human-readable source code.<sup>10</sup> It is usually done in two steps: initially, an intermediate copy of the object code to be decompiled is made; next, the copy is translated into the desired form of source code.

There are three common uses of decompilation which serve to define three different types of decompilation:

(1) in studying the underlying concepts of a work and using the results to create another functionally equivalent work: a process often called 'reverse-engineering';<sup>11</sup>

(2) in obtaining the information necessary for communicating with the decompiled program in the course of developing another computer program: that is, in technical parlance, for achieving the 'interoperability' between the two programs;

(3) similar to the second use but for achieving interoperability with *any* program, including programs other than the one decompiled.

These three uses will respectively be referred to as 'decompilation-for-reverse-engineering,' 'decompilation-for-specific-interoperability,' and 'decompilation-for-general-interoperability.' While decompilation-for-reverse-engineering is clearly for a competing purpose, decompilation-for-specific-interoperability is not because it does not seek to replace the decompiled program in the market. On the other hand, decompilation-for-general-interoperability may or may not be for a competing purpose, depending on whether or not it is aimed at producing a rival program.

<sup>9</sup> Committee of Experts on a Possible Protocol to the Berne Convention for the Protection of Literary and Artistic Works, para 37: [1992] Copyright 30, 35.

<sup>10</sup> Expressed in, for instance, an assembly language or a high-level language such as Basic, Pascal, or C.

<sup>11</sup> One technical definition of reverse-engineering is 'the act of creating a set of functional specifications for a system by someone other than the original designer based on an analysis of an existing system': Pamela Samuelson, 'Reverse-Engineering Someone Else's Software: Is it Legal?' (Jan 1990) IEEE Software 90, 91.

*Status of decompilation in Hong Kong*

The issue of decompilation has not been litigated before any UK or Hong Kong court. But since decompilation necessarily involves the intermediate copying of the whole of a program's object code, it violates the copyright owner's exclusive right of reproduction of his work under the current copyright law.<sup>12</sup> Furthermore, because it involves translating object code into source code, it also infringes the owner's exclusive right of making an adaptation.<sup>13</sup> In the same vein, to the extent that reverse-engineering involves decompilation, the former will also fall foul of the current copyright law.<sup>14</sup>

## Fair dealing defence

At first glance, the fair dealing defence under s 6 of the Copyright Act of 1956 may seem to be of some assistance to a decompiler in Hong Kong. Under that section, no 'fair dealing' with a literary work (which includes computer programs)<sup>15</sup> will constitute an infringement of the copyright in the work if it is for 'research or private study,' 'criticism or review,' or 'reporting current events.' Among these legitimate purposes, 'research or private study' is most relevant to decompilation. In the provision, 'research' is unqualified and presumably includes both private and commercial research.<sup>16</sup> However, one must observe that the defence requires two elements: (1) the dealing must be 'fair'; and (2) it must be for the specific purpose of 'research or private study.' These two elements are not entirely independent, as is evident from a remark on fair dealing by Lord Denning MR in *Hubbard v Vosper*:

You must consider first the number and extent of the quotations and extracts. Are they altogether too many and too long to be fair? Then you must consider the use made of them. If they are used as a basis for comment, criticism or review, that may be a fair dealing. If they are used to convey the same information as the author, *for a rival purpose*, they may be *unfair*. Next, you must consider the proportions. To take long extracts and attach short comments may be unfair. But short extracts and long comments may be fair. Other considerations may come to mind also. But, after all is said and done, it must be a matter of impression.<sup>17</sup>

<sup>12</sup> Copyright Act 1956, s 2(5)(a) as supplemented by Copyright (Computer Software) Amendment Act 1985, s 2.

<sup>13</sup> Copyright Act 1956, s 2(5)(f) as supplemented by Copyright (Computer Software) Amendment Act 1985, s 1(2).

<sup>14</sup> Note that decompilation is often an integral part of reverse-engineering because it is easier to extract the underlying concepts of a program from the source code than the object code.

<sup>15</sup> By virtue of Copyright (Computer Software) Amendment Act 1985, s 1(1).

<sup>16</sup> Otherwise the phrase should have been 'private research or study' instead of 'research or private study.' This view is also in line with that of the LRC: LRC report, para 6.31.

<sup>17</sup> [1972] 2 QB 84, 94 (emphasis supplied).

In other words, when assessing whether a dealing is 'fair,' the court must consider, *inter alia*, the purpose and nature of the dealing. Unfortunately not much light has been shed by later cases on this point, despite the potentially wide applicability of the defence. But it should be clear that if 'research or private study' is only a pretext and the main purpose is to adversely affect the economic interest of the copyright owner, it may militate against a finding of fair dealing. A similar conclusion can be drawn when the legitimate purpose is accompanied by other purposes.

Hence it seems unlikely that competing decompilation would be afforded a defence of fair dealing in Hong Kong. The defence would fail on either one (or both) of the following two grounds: (1) the purpose is not for 'research or private study' (or any of the other legitimate purposes) but for developing a competing product; and (2) the dealing is not 'fair.'

This conclusion is buttressed by an observation of the newly enacted Layout-Design (Topography) of Integrated Circuits Ordinance.<sup>18</sup> The ordinance contains provisions which expressly permit reproductions of a layout-design done 'for the sole purpose of evaluation, analysis, research or teaching,'<sup>19</sup> and using the results of the analysis to 'create a different layout-design (topography).'<sup>20</sup> This deliberate effort of allowing reproductions for the creation of a competing semiconductor chip suggests that the current copyright law is not perceived as allowing reproductions for the purpose of developing a competing product, and thus it is deemed necessary to introduce a form of *sui generis* copyright regime for semiconductor chips.

It is less clear if non-competing decompilation would have the same fate. But there are two grounds to suggest that the fair dealing defence would also not be available to such decompilation under the current law. First, although it might be easier to argue in this case that the purpose is within that of 'research or private study,' it would still be difficult to claim that the intermediate copying and the subsequent translation of the *whole* of a computer program's object code are 'fair.' Second, except for a recommendation to extend the fair dealing defence to all copyright works,<sup>21</sup> the LRC report contains no suggestion that the defence should deviate from the existing law.<sup>22</sup> At the same time, the report has firmly rejected any limited right to decompilation on the ground that it is 'tantamount to creating another exception to infringement.'<sup>23</sup> This carries the negative implication that, at least in the opinion of the LRC, decompilation *per se* is not a permitted act under the current law, nor will it be under the LRC's proposed reform.

<sup>18</sup> Ordinance No 17 of 1994, enacted on 30 March 1994.

<sup>19</sup> s 5(c).

<sup>20</sup> s 5(d).

<sup>21</sup> LRC report, paras 6.33-4.

<sup>22</sup> This is reinforced by a recommendation that no statutory definition be given to fair dealing: para 6.30.

<sup>23</sup> Note 6 above.

To a large extent, Hong Kong's prohibition of decompilation does not accord with the laws of its two major trading partners, the US and the European Community, both of which permit some form of decompilation as an exception to copyright infringement. This will become apparent as one looks at the status of decompilation in these two jurisdictions.

### Decompilation in the US

Section 117 of the US Copyright Act of 1976<sup>24</sup> allows the owner of a copy of a computer program to make another copy or adaptation of that program. But the right is subject to conditions, namely, that the new copy or adaptation must be 'created as an essential step in the utilization of the computer program in conjunction with a machine and that it is used in no other manner,' or is 'for archival purposes only.' Since the intermediate copying and subsequent translation in a decompilation invariably do not satisfy these conditions, the provision is regarded by some as denying a general right to decompile.<sup>25</sup>

On the other hand, the 1976 Act also embodies a 'fair use' doctrine which is capable of being a defence of decompilation. Section 107 states that the 'fair use' of a copyrighted work 'for purposes such as criticism, comment, news reporting, teaching (including multiple copies for classroom use), scholarship, or research' is not an infringement of copyright. Unlike the 'fair dealing' defence in Hong Kong, the legitimate purposes for fair use listed in s 107 are inclusive rather than exhaustive. Thus there is no principled basis for holding that decompilation may never be a fair use under the US copyright law.

Because of the co-existence of s 117 and s 107, the status of decompilation in the US had not been entirely clear. It was only in two recent cases, *Atari Games v Nintendo*<sup>26</sup> and *Sega Enterprises v Accolade*,<sup>27</sup> that the US courts were asked to address the issue. The facts in the two cases are similar: both involved certain digital 'lock and key' systems which the plaintiffs (Nintendo and Sega, both manufacturers of home video game consoles) employed to prevent unlicensed game cartridges to play on their consoles; and in both, the defendants (Atari and Accolade, both manufacturers of game cartridges) had decompiled the object codes in the respective plaintiffs' game cartridges to find out the electronic signals necessary for 'unlocking' the plaintiffs' consoles. In both cases, the district courts had issued preliminary injunctions against the defendants, but the appellate courts were willing to confer a limited privilege on decompilation based on the fair use doctrine.

<sup>24</sup> Title 17, USC.

<sup>25</sup> See, eg, Arthur R Miller, 'Copyright Protection for Computer Programs, Databases, and Computer-Generated Works: Is Anything New Since CONTU?' (1993) 106 Harvard LR 977, 1014.

<sup>26</sup> 975 F 2d 832 (Fed Cir 1992). See also the decision on remand and the subsequent orders (ND Calif 15 April and 17 May 1993).

<sup>27</sup> 977 F 2d 1510 (9th Cir 1992).

*The Sega decision*

Despite the different outcomes,<sup>28</sup> *Sega* and *Atari* are consistent in most respects. Of the two, *Sega* gives the more thorough analysis of fair use. There the Court of Appeals for the Ninth Circuit held that Accolade's intermediate copying of the object code from a Sega video game cartridge as a preliminary step to decompiling the code was, prima facie, copyright infringement.<sup>29</sup> However, it was held that fair use will permit decompilation under certain circumstances. What merits detailed attention in the case is the court's analysis of whether Accolade's decompilation of Sega's object code was fair use. Reference was made to the four factors listed in s 107 of the 1976 Act: (1) the purpose and character of the use, including whether such use is of a commercial nature or is for nonprofit educational purposes; (2) the nature of the copyrighted work; (3) the amount and substantiality of the portion used in relation to the copyrighted work as a whole; and (4) the effect of the use upon the potential market for or value of the copyrighted work.

Although the third factor weighed heavily against Accolade's intermediate copying of the whole of Sega's object code, the court found the other three factors in favour of fair use.<sup>30</sup> First, despite the commercial aspect of Accolade's decompilation,<sup>31</sup> the court regarded its main purpose as identifying the functional requirements for Accolade's independently created cartridges in order to be compatible with the Sega console. This was held to be a legitimate purpose because it had led to 'an increase in the number of independently designed video game programs offered for use with the [Sega] console,' which is precisely the kind of 'growth in creative expression' 'that the Copyright Act was intended to promote.'<sup>32</sup>

Second, the court discussed the nature of Sega's copyrighted work and concluded that computer programs 'are, in essence, utilitarian articles.'<sup>33</sup> Observing that computer programs often 'contain unprotected aspects that cannot be examined without copying,'<sup>34</sup> the court pronounced that they should receive 'a lower degree of protection than the more traditional literary works.'<sup>35</sup>

<sup>28</sup> The district court's judgment in *Sega* was reversed on appeal. But in *Atari*, the facts were complicated by allegations that Atari had made false representations to the US Copyright Office and obtained a copy of Nintendo's source program deposited there. Partly in consideration of this illegal act, the district court's preliminary injunction was upheld by the Court of Appeals for the Federal Circuit.

<sup>29</sup> Contrary to the US Copyright Act of 1976, s 106(1).

<sup>30</sup> The court gave very little weight to the third factor because only a limited portion of Sega's code had been used in Accolade's final product (note 27 above, at p 1526). Note that this is not the only instance where a US court accepts the copying of the whole of a work as 'fair use.' Another well-known decision is *Sony v Universal City Studios*, 464 US 417 (1984) in which it was held that home video-taping the whole of a television programme for time-shifting purpose is fair use.

<sup>31</sup> Copying for a commercial purpose is presumed to be unfair: *Sony*, *ibid*; *Harper & Row, Publishers v Nation Enterprises*, 471 US 539 (1985).

<sup>32</sup> 977 F 2d 1510, 1523.

<sup>33</sup> *Ibid*, p 1524.

<sup>34</sup> *Ibid*, p 1526.

<sup>35</sup> *Ibid*. The court's reasoning was based on s 102(b) of the 1976 Act. 'If disassembly of copyrighted object code is per se an unfair use, the owner of the copyright gains a de facto monopoly over the functional aspects of his work aspects that were expressly denied copyright protection by Congress' (*ibid*).

Third, the court noted that the interface information was acquired by Accolade for the purpose of achieving interoperability with the Sega console and not for developing substitute products for the Sega cartridges. Hence Accolade's independently created cartridges did not usurp the market for Sega's cartridges but merely competed with them. This competitiveness in the market is what the Copyright Act aims to maintain.<sup>36</sup>

#### *Impact of the Sega and Atari decisions*

Both *Sega* and *Atari* have sent a clear message that decompilation of a copyrighted program — with the intermediate copying and subsequent adaptation it entails — will be permitted as fair use in the US if: (1) it is necessary for gaining access to unprotected elements of the program; and (2) gaining such access is for creating another product which is not a substitute for the program. This would exclude competing decompilation from fair use, but would admit non-competing decompilation that is necessary for achieving interoperability, whether specific or general.<sup>37</sup>

#### *The NEC decision*

In spite of *Sega* and *Atari*, the question whether there is in the US a complete ban on competing decompilation is far from settled. In an earlier case, *NEC v Intel*,<sup>38</sup> the court seemed to permit competing decompilation, albeit on grounds other than fair use. NEC had admittedly decompiled Intel's microcode in a chip, and from it 'derived' the microcode in NEC's functionally equivalent chip. But while the first version of NEC's microcode was very similar to Intel's, the final version as incorporated in NEC's chip was not. Probably because NEC had adopted a 'clean room' procedure,<sup>39</sup> the court ignored the intermediate steps in NEC's initial decompilation and the fact that NEC's final version was a 'derivative work' of Intel's microcode.<sup>40</sup> Judging solely on NEC's final version, the court did not find any substantial similarity necessary for copyright infringement.<sup>41</sup>

<sup>36</sup> '[Sega's] attempt to monopolize the market by making it impossible for others to compete runs counter to the statutory purpose of promoting creative expression and cannot constitute a strong equitable basis for resisting the invocation of the fair use doctrine' (ibid, p 1524).

<sup>37</sup> Although in *Sega* the decompilation in question was only for specific interoperability, the decision tacitly supports the view that decompilation-for-general-interoperability which is non-competing is also fair use.

<sup>38</sup> 10 USPQ 1177 (1989).

<sup>39</sup> This is a technique in reverse-engineering that attempts to avoid copyright infringement. It involves two teams of programmers: the first examines the original program, often by means of decompilation, and extracts its functional specifications; the second works from the specifications and performs an independent implementation.

<sup>40</sup> Contrary to the US Copyright Act of 1976, s 106(2).

<sup>41</sup> In addition, the court held that in view of the hardware constraints involved, copyright protection could only be against virtually identical copying. This may be viewed as an application of the US 'merger doctrine,' which states that where an idea can be expressed in only one way, that expression will not be protected: *Landsberg v Scrabble Crossword Game Players*, 736 F 2d 485 (9th Cir 1984).

If NEC is still good law, it would mean that if a decompiler has adopted a clean room procedure, a US court would only find infringement when the results of decompilation have been used to create a program substantially similar to the one decompiled.<sup>42</sup> This position on decompilation is even more liberal than that in *Sega* and *Atari*.<sup>43</sup>

### Decompilation in the European Community

In an effort to harmonise software protection within the Community, the EC Council adopted in May 1991 a Directive on the Legal Protection of Computer Programs ('Software Directive').<sup>44</sup> The debate during the drafting of the Directive was centred on one issue: decompilation.<sup>45</sup>

Involved in the debate were two camps. On one side was the Software Action Group for Europe (SAGE) representing the interests of large US firms such as IBM, DEC, Microsoft, and Lotus. On the other was the European Committee for Interoperable Systems (ECIS) representing the interests of European and Japanese firms and some small software houses. Because of the disparate positions in the EC market, the ECIS firms found themselves unable to compete with the SAGE firms directly by producing rival hardware, and could only thrive on producing hardware and software that were compatible with the latter's products. Thus not surprisingly, the SAGE strongly opposed, whereas the ECIS fervently fought for, an express right to decompile.

There were two main strands to the SAGE arguments: (1) the decompilation right was unnecessary as manufacturers would often have made available, through published materials or upon request, sufficient information to enable others to develop programs that were interoperable with theirs; and (2) introducing the right would be changing the fundamental principles of copyright to favour imitators and reduce reward for innovation. On the other hand, the ECIS maintained two main contentions: (1) the absence of any express

<sup>42</sup> This is the view held by Samuelson (note 11 above), p 96.

<sup>43</sup> A US court in another case even accepted decompilation on the ground that it was a 'standard practice' in the industry. 'The mere fact that defendant's engineers dumped, flowcharted, and analyzed plaintiff's code does not, in and of itself, establish pirating. As both parties' witness admitted, dumping and analyzing competitors' codes is a standard practice in the industry': *EF Johnson v Uniden*, 623 F Supp 1485 (D Minn 1985). But this seems to be a minority view in the US.

<sup>44</sup> At the time of adopting the Directive, the laws of the EC member states on software protection differed widely. For instance, the term of software protection was only 25 years under French law whereas it was the life of author plus 50 years under the UK law; and while the UK standard of originality for software was low, the German standard was akin to the high standard for novelty in patent law.

<sup>45</sup> For a more detailed account of the background to the Directive and the controversy over decompilation, see Ian A Staines, 'The European Commission's Proposal for a Council Directive on the Legal Protection of Computer Programs' [1989] 6 EIPR 183; W R Cornish, 'Inter-operable Systems and Copyright' [1989] 11 EIPR 391; William T Lake, John H Harwood II, and Thomas P Olson, 'Seeking Compatibility or Avoiding Development Costs? A Reply on Software Copyright in the EC' [1989] 12 EIPR 431; Michel Colombe and Caroline Meyer, 'Seeking Interoperability: An Industry Response' [1990] 3 EIPR 79; Michel Colombe and Caroline Meyer, 'Interoperability Still Threatened by EC Software Directive: A Status Report' [1990] 9 EIPR 325.



right to decompile would merely strengthen the already dominant positions of the major corporations and impede competition; and (2) the prohibition of decompilation would be equivalent to using copyright to protect ideas in programs.

Article 6 of the Software Directive is the result of this debate. Allowing decompilation only under limited conditions, the carefully worded provision reflects the EC Council's view of how the proper balance between the copyright owner and the user<sup>46</sup> of a computer program should be drawn. Article 6.1 states that the authorisation of the rightholder is not required where reproduction of a program's code and translation of its form are 'indispensable to obtain the information necessary to achieve the interoperability of an independently created computer program with other programs,' provided that three conditions are satisfied: (a) the acts are performed by the licensee or by another person having a right to use a copy of a program, or on their behalf by a person authorized to do so; (b) the information necessary to achieve interoperability has not previously been readily available to the persons referred to in subparagraph (a); and (c) these acts are confined to the parts of the original program which are necessary to achieve interoperability.

This article makes it clear that the only legitimate purpose for decompilation without authorisation is to achieve interoperability with other programs. Even maintenance, by itself, is not a reason justifying decompilation.<sup>47</sup> Moreover, due to condition (b), decompilation is only allowed when the information needed for the legitimate purpose 'has not previously been readily available.' It follows that if the information is obtainable through observing, studying, or testing the program and extracting its underlying ideas and principles by legal means,<sup>48</sup> decompilation will not be allowed. This gives the rightholder strong incentives to disclose the information if he wants to prevent his program from being decompiled.

Another important effect of article 6.1 is that it sanctions not only decompilation-for-specific-interoperability but also decompilation-for-general-interoperability. This is because the provision only refers to the interoperability of an independently created program 'with other programs.' Hence the new program may be for a competing purpose. This point is clearly set out in the Communication of the EC Council's Common Position to the European Parliament:

<sup>46</sup> Who may be the author of a rival program.

<sup>47</sup> But maintenance will be an exception to the restricted acts under Art 5.1 of the Directive.

<sup>48</sup> As permitted under art 5.3 of the Directive, which reads: 'The person having a right to use a copy of a computer program shall be entitled, without the authorization of the rightholder, to observe, study or test the functioning of the program in order to determine the ideas and principles which underlie any element of the program if he does so while performing any of the acts of loading, displaying, running, transmitting or storing the program which he is entitled to do.'

Decompilation is permitted by Article 6 to the extent necessary to ensure the interoperability of an independent created computer program. Such a program may connect to the program subject to decompilation. *Alternatively, it may compete with the decompiled program* and in such circumstances will not necessarily connect to it.<sup>49</sup>

To prevent any misuse of the information lawfully obtained by decompilation, article 6.2 stipulates that such information is not to be: (a) used for goals other than to achieve the interoperability of the independently created computer program; (b) given to others, except when necessary for the interoperability of the independently created computer program; or (c) used for the development, production, or marketing of a computer program substantially similar in its expression, or for any other act which infringes copyright. These three restrictions seek to ensure that decompilation is not used as a mere pretext. Due to restriction (a), decompilation-for-reverse-engineering is clearly excluded.

Article 9.1 of the Directive expressly states that any contractual provisions contrary to article 6 'shall be null and void.' This ensures that legitimate decompilation will not be excluded by any user licence, particularly when the parties to the licence are not of equal bargaining power. Not only is this another indication of the EC Council's strong support for a limited right to decompilation, it is also a manifestation of the competition rules in the EC.<sup>50</sup>

### Right to decompilation in Hong Kong?

To appreciate Hong Kong's stance on decompilation from an international perspective, it is instructive to compare it with those of the US and the EC. Table 1 on page 25 summarises the result of this comparison. It is clear that among the three jurisdictions, Hong Kong is the most restrictive on the right to decompilation.

<sup>49</sup> SEC 87 Final SYN 183, 18 January 1991, para 4.7. (emphasis supplied.)

<sup>50</sup> In particular art 85 (prohibition against anti-competitive agreements) and art 86 (prohibition against abuse of a dominant position) of the Treaty of Rome.

**Table 1** Status of Decompilation

Type of Decompilation	Hong Kong	US	EC
Decompilation-for-reverse-engineering	Not permitted	Position not clear	Not permitted
Decompilation-for-specific-interoperability	Not permitted	Permitted	Permitted
Decompilation-for-general-interoperability	Not permitted	Permitted if product is non-competing; position not clear if product is competing	Permitted whether or not product is competing

*Arguments against decompilation*

As the issue of decompilation has not attracted much debate in Hong Kong, it is worth looking at the arguments against a decompilation right that have been raised in other jurisdictions.<sup>51</sup> Those arguments that are relevant to Hong Kong can be classified into four categories: (i) the 'fundamental principle' argument, (ii) the 'fairness' argument, (iii) the 'trade secret' argument, and (iv) the 'market force' argument. These arguments will be considered in the context of Hong Kong.

*Fundamental principle argument*

The 'fundamental principle' proponents argue that since the legislature has decided to protect computer software as literary work and rejected a sui generis regime, it is the intention of the legislature to give the same treatment to computer software as any other type of literary works. They contend that just as a novel written in an obscure foreign language unfamiliar to the ordinary reader cannot be copied or translated without the copyright owner's consent, an object code, no matter how unintelligible it may be to the ordinary user, should enjoy the same level of protection.

<sup>51</sup> See, eg, Miller (note 25 above); Lake, Harwood, and Olson (note 45 above).

The obvious answer to the 'fundamental principle' argument is that it overlooks the fact that even within the genre of traditional literary works, the courts have never applied the same treatment across the board. Copyright protection for biographies and scientific and historical works, for instance, has always been 'thinner' than that for novels and plays: the protection for the former group of works is usually only against verbatim copying, whereas the protection for the latter group often casts a much wider net and prohibits also the copying of plots.<sup>52</sup> Thus even if software is to be regarded as no different from these traditional literary works, there is still the question whether it is more analogous to novels and plays or to other types of literary works.

A more fundamental flaw in the 'fundamental principle' argument is its failure to appreciate the nature of computer programs. Unlike traditional literary works such as a book, where a person can find out its underlying ideas and principles by reading it, an object code does not generally reveal its underlying ideas and principles to the person who reads the code — even if he is a skilled programmer — much less to the person who runs the code.<sup>53</sup> The analogy between an object code and a novel written in an obscure foreign language is erroneous in at least two respects. First, by its nature, a novel is created for reading, whereas an object code is for execution in a computer. Second, in the case of the incomprehensible novel, there is a strong argument, based on the doctrine in *British Leyland Motor v Armstrong Patents*,<sup>54</sup> that the purchaser of the novel has an implied right to make a private translation of the work in order to read it, at least in circumstances where a translation is not already available. But for an object code, the purchaser is unlikely to have such an implied translation right since no translation is necessary for running the code.<sup>55</sup>

Thus ironically, in pursuing the objective of providing equal treatment to all literary works, the 'fundamental principle' proponents are actually putting the purchaser of a computer program in a disadvantaged position compared to the purchaser of a novel. This self-contradiction is an inevitable result of their narrow and overly simplistic view of copyright law.

<sup>52</sup> See, eg, *Corelli v Gray* (1913) 30 TLR 116; *Fernald v Jay Lewis* [1975] FSR 499; *Ravenscroft v Herbert* [1980] RPC 193.

<sup>53</sup> This is one reason why copyright is still viewed by some as unsuitable for software protection: see, eg, Richard H Stern, 'The Paperback Case: Part 3, Misconceptions About Functionality' (Feb 1991) *IEEE Micro* 48; Zheng Chengsi and Michael Pendleton, *Copyright Law in China* (Sydney: CCH International, 1991), pp 194–200.

<sup>54</sup> [1986] FSR 221. The House of Lords' application of the principle of 'non-derogation from grant' to spare parts in the case can be generalized as a doctrine that a purchaser, in acquiring a copy of a work, acquires also the inherent right to perform acts necessary for using the work. Furthermore, the purchaser cannot be deprived of this inherent right by the copyright owner of the work.

<sup>55</sup> Indeed object code is precisely the form in which a computer program may be run.

### Fairness argument

The 'fairness' argument contends that permitting decompilation allows a second comer to reap the fruits of a successful program in whose development the owner has incurred much risk and expense. This is regarded as extremely unfair to the program copyright owner, given that the cost of developing a computer program is so high.<sup>56</sup> If an exemption for decompilation is granted, the argument goes, any incentive to produce innovative works will be eroded, defeating one of the basic purposes of copyright law.

While the 'fairness' argument is not entirely without force, it is too broad-brushed by not distinguishing between different types of decompilation. Among the three types of decompilation described earlier, the only one that may come close to being described as 'unfair' is decompilation-for-reverse-engineering. But even there opinions differ. There are some who consider this type of decompilation as equivalent to the kind of analysis allowed in patent law,<sup>57</sup> and since copyright accords lesser protection to the proprietor than patent, such decompilation should a fortiori be allowed under copyright law.<sup>58</sup> There are others who argue that decompilation-for-reverse-engineering has the merit of enabling competitors to create better programs, hence the question of copyright infringement should not depend on any incidental copying but on whether the new program has taken the protectable expressions of the decompiled program.<sup>59</sup>

It is admitted that the question whether decompilation-for-reverse-engineering should be permitted is a difficult one to answer. But even if one concedes that such decompilation is 'unfair,' one cannot say that decompilation-for-specific-interoperability is in any way 'unfair.' Nor is decompilation-for-general-interoperability 'unfair' if it is not for a competing purpose. Indeed, if these types of decompilation are used for developing new and innovative works, they should be encouraged as such creative efforts are exactly the kind of activities that copyright law aims to stimulate.

The only question that warrants further consideration is whether decompilation-for-general-interoperability aimed at developing a competing product should be allowed. Here one only has to draw on the experience of the EC to find the answer. To help their software industry battle against foreign domination, particularly that of the US, the EC has decided to allow decompilation-for-general-interoperability even if it is for a competing purpose. Considering that our local software firms are in a still weaker position against the major foreign corporations, it is even more important for Hong Kong to sanction such decompilation. Otherwise, we are only destroying our software industry by hamstringing our own software firms.

<sup>56</sup> See, eg, Miller (note 25 above), p 1026.

<sup>57</sup> See, eg, UK Patents Act 1977, s 60(5)(b).

<sup>58</sup> See, eg, Cornish (note 45 above), p 392.

<sup>59</sup> See, eg, Samuelson (note 11 above), p 96. This view is in essence the ratio of the NEC decision in the US (note 38 above).

### Trade secret argument

The 'trade secret' proposition is a relatively simple one. Its adherents assert that the most valuable asset in a computer program is its underlying ideas and principles, which are often its trade secrets. They claim that if decompilation is permitted, the protection of such trade secrets will be severely impaired.<sup>60</sup>

The response to this assertion is also a simple one. One only needs to point out that there is no general principle in intellectual property law which prevents the underlying ideas of a work from being studied. Copyright does not protect ideas.<sup>61</sup> Patent law goes even further and compels the proprietor to disclose his ideas to the public in return for a monopoly on his invention.<sup>62</sup> And although the law of confidence protects trade secrets against their recipients, it does not forbid anyone else from finding out the secret by legal means, such as reverse-engineering.<sup>63</sup> Indeed, when a manufacturer puts his product onto the market, the law will only protect the trade secret embedded in his product for the period of the 'lead time,' that is, the time required for his competitors to discover the secret by studying, testing, and analysing his product. This principle is the basis of the well-known 'spring-board' doctrine<sup>64</sup> in the law of confidence.

Much of the fear of the 'trade secret' proponents lies in their unfounded assumption that, compared with the reverse-engineering of other industrial products, decompilation of software reduces the lead time much more drastically.<sup>65</sup> The truth is, even for a skilled programmer, decompilation is a tedious and laborious process in which few pirates would be willing to invest their time.<sup>66</sup> Furthermore, to avoid copyright infringement, the decompiler must expend additional effort in producing independent expressions of the ideas revealed by the decompilation. These considerations and those above have eliminated much of the force in the 'trade secret' argument.

### Market force argument

The 'market force' argument asserts that it is unnecessary to grant a statutory licence to decompilation. It contends that manufacturers of hardware and software have often released technical information to application program

<sup>60</sup> See, eg, Miller (note 25 above), p 1026.

<sup>61</sup> Although not enshrined in any UK statute, the idea/expression dichotomy is recognised and applied by the courts: see, eg, *Ladbroke (Football) v William Hill (Football)* [1964] 1 WLR 273; *LB (Plastics) v Swish Products* [1979] RPC 551.

<sup>62</sup> See, eg, UK Patents Act 1977, ss 14(3), 72(1)(c).

<sup>63</sup> Thus it is perfectly legal for anyone to find out the secret formula for a soft drink by chemical analysis and use the same formula to develop a rival soft drink. But the drink must be marketed under a different trademark.

<sup>64</sup> The doctrine states that a recipient of confidential information is not allowed to use the information prior to the expiry of the lead time, so as to strip him of any unfair advantage he may have over his competitors: *Terrapin v Builders' Supply* [1967] RPC 375.

<sup>65</sup> See, eg, Miller (note 25 above), p 1026.

<sup>66</sup> That is why the most common form of copyright infringement everywhere in the world is still by slavish copying alone.

designers for commercial reasons, particularly to encourage third parties to write programs that are interoperable with their products.<sup>67</sup> This is crucial to the manufacturer's securing the largest possible share of the market. According to the 'market force' proponents, those manufacturers who refuse to disclose their technical information will have little chance of success in the long run. Thus, these proponents claim, the market force has already provided the best answer for promoting interoperability.

The primary response to this argument is that it is naïve to believe that the already dominant players in the computing industry will be lured by their concern over market shares to voluntarily release their technical information. The reality is exactly the opposite: unless there is some other compelling force, the dominant corporations will try to withhold as much information as possible from the public to maintain their dominance in the market. This they can easily accomplish by, inter alia, making their user manuals incomplete or ambiguous to those who want to develop products that are interoperable with theirs.

The history of the computing industry contains numerous examples contradicting the 'market force' theory. One good illustration is the 1984 *IBM Settlement* case in the EC.<sup>68</sup> The case arose from complaints lodged with the EC Commission against IBM for its refusal to release the interface information prior to its marketing the System/370 computer. By withholding the information before announcing the new computer, IBM was able to monopolise the market for several months after the product was put onto the market by acting as the sole supplier of compatible peripherals. The Commission investigated IBM's alleged abuse of its dominant position<sup>69</sup> and IBM finally gave an undertaking to disclose the relevant interface information at least four months prior to launching a new product. IBM did so not because it was worried about its market share, but because it was concerned with the possible proceedings for its infringement of the EC competition rules. This illuminating example demonstrates that in certain circumstances, it is the law that is more effective than the market force in harnessing the potential monopolists.

On the other hand, even if the 'market force' theory is correct, there is no harm in giving legal recognition to the market force. Indeed a well drafted provision may assist the market force by adding a legal push. A good example is article 6.1 of the EC Software Directive discussed earlier.<sup>70</sup> By giving the lawful user of a program a right to decompilation where 'the information necessary to achieve interoperability has not previously been readily available,' the article in effect compels the program copyright owner to disclose such information if he wants to prevent others from decompiling his program. This

<sup>67</sup> See, eg, Lake, Harwood and Olson (note 45 above), p 431; Miller (note 25 above), p 1031. It seems that the LRC also subscribe to this view: LRC report, para 13.32.

<sup>68</sup> [1984] 3 CMLR 147; see also Bulletin of the EC 10-1984, p 96.

<sup>69</sup> Contrary to Art 86 of the Treaty of Rome.

<sup>70</sup> See above, 'Decompilation in the EC.'

is far more effective in promoting the dissemination of information than having to wait for the market to exert its force on the copyright owner.

### Simplicity argument of the LRC

Not included in the four categories above is the LRC's 'simplicity' argument. It contends that the copyright system should be kept simple, and so decompilation should not be allowed as an 'another exception to infringement.' As a basic principle this goal of pursuing simplicity is sound, but copyright law has always been complex from the outset due to its wide range of subject matter. Hence exceptions are inevitable and they necessarily vary from one type of work to another.<sup>71</sup> And the gravest mistake that our legislature can make is to inject an artificial simplicity into the law at the expense of the interests of the weaker parties concerned, such as our local software firms vis-à-vis the giant US firms.

### Conclusion

However formidable they may look, the arguments against a decompilation right do not really stand up to scrutiny. At the root of all the opposition lies the fear that a right to decompilation may undermine the protection of computer software. Such fear is unfounded as its genesis can invariably be traced back to some fundamental flaws, such as: a failure to distinguish different uses of decompilation; an inability to appreciate the nature of software; an insufficient understanding of the copyright law; a blind faith in the market force; and sometimes a misconception of intellectual property law altogether. At its strongest, this opposition provides only an arguable case against decompilation-for-reverse-engineering. It cannot be pleaded against decompilation-for-specific-interoperability or decompilation-for-general-interoperability at all.

The issue of a decompilation right is essentially a question of finding the right balance between the user and the copyright owner of a computer program, taking into account the interest of the society as a whole. In sharp contrast with their counterparts in the US and the EC, software firms in Hong Kong presently do not have any right to decompile a program, no matter how legitimate their needs are in accessing the ideas underlying the program. This is so even when decompilation is the only means of obtaining the necessary information for communicating with the program. The damaging effect of this on these firms, especially those whose business depends on their abilities to develop products compatible with existing programs, is just too obvious to be overlooked.

Although the question whether decompilation-for-reverse-engineering should be permitted remains a difficult issue, decompilation-for-specific-interoperability and decompilation-for-general-interoperability (whether or

<sup>71</sup> See, eg, the exceptions to infringement under the UK Copyright Act 1956, ss 6-10; and the more recent Copyright, Designs and Patents Act 1988, ss 28-76.



not for a competing purpose) should be allowed in Hong Kong. In this connection, the newly enacted s 50B of the UK Copyright, Designs and Patents Act 1988,<sup>72</sup> which provision represents UK's compliance with the EC Software Directive, is an excellent model for Hong Kong to follow. Contrary to the view of the LRC, this UK provision expressly confers upon the lawful user of a computer program a decompilation right. But the right is exercisable only for the purpose of achieving interoperability, as stipulated by the conditions in s 50B(2). That is, it is exercisable only if:

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

Condition (a) makes it clear that both decompilation-for-specific-interoperability and decompilation-for-general-interoperability are permitted, regardless of whether the final product competes with the decompiled program. Further restrictions on the decompilation right are imposed by s 50B(3). In particular, the conditions in s 50B(2) are not met if the lawful user:

- (a) has readily available to him the information necessary to achieve the permitted objective;
- (b) does not confine the decompiling to such acts as are necessary to achieve the permitted objective;
- (c) supplies the information obtained by the decompiling to any person to whom it is not necessary to supply it in order to achieve the permitted objective; or
- (d) uses the information to create a program which is substantially similar in its expression to the program decompiled or to do any act restricted by copyright.

It is submitted that s 50B strikes a fair balance between the copyright owner and the user of a computer program, and should be adopted in Hong Kong. By granting the user a right to decompilation when the act is necessary for achieving interoperability, this provision will ensure that our local software industry is given the opportunity to feed on the state-of-the-art technologies.

<sup>72</sup> Pursuant to the UK Copyright (Computer Programs) Regulations 1992, No 3233 of 1992 enacted on 16 December 1992. The new provision came into force on 1 January 1993.

At the same time, the conditions imposed by the provision on the decompiler will protect the interest of the copyright owner — often a dominant software firm — against unfair competition and possible abuses of the decompilation right. Furthermore, by adopting the provision, Hong Kong will be on a par with its trading partners in the EC and the US.

This is not to say that Hong Kong should follow every legal step of its trading partners. But quite apart from the theoretical arguments, there is a very real and practical consideration for Hong Kong, namely, that it cannot afford not to have a decompilation right. To appreciate this, one only needs to look at the dire consequence of the absence of such a right in Hong Kong: that while software firms in the US and the EC have a right to decompile programs originating from Hong Kong, our local software firms are denied a reciprocal privilege by our own law. There is neither justification for nor wisdom in this. And until a decompilation right is firmly established in Hong Kong, this situation will continue to cripple the competitiveness of our own software industry.

This situation must be changed, and the time to act is now.

K H Pun\*

## Dismissing Academics

### Introduction

Unravelling the rights and obligations which form the basis of any employment relationship is not always a simple task. Such an undertaking is made no less easy where an employment contract comprises terms, some of which are statutorily derived, and decisions about which, are the subject-matter of judicial review. In such circumstances, it is perhaps no small wonder that a dispute involving a law lecturer and the University of Hong Kong managed to last the full distance by starting off in the High Court,<sup>1</sup> going on appeal to the Court of Appeal,<sup>2</sup> and finally making it to the Privy Council.<sup>3</sup> As if to deride the notion of finality, the case has continued in draw publicity long after the Law Lords gave their unanimous decision on the matter.<sup>4</sup>

\* Lecturer, Department of Computer Science, University of Hong Kong.

<sup>1</sup> *Spruce, In the matter of an application by Jill Spruce for Judicial Review*, High Court, Miscellaneous Proceedings No. 2332 of 1990.

<sup>2</sup> *Spruce, Jill, v The University of Hong Kong* [1991] 2 HKLR 444.

<sup>3</sup> *Spruce, Jill, v The University of Hong Kong*, Privy Council, Appeal No. 27/92.

<sup>4</sup> See: 'University seeks \$3m,' *South China Morning Post*, 7 January 1994; 'Bid to halt bankruptcy move,' *South China Morning Post*, 25 April 1994; 'Top judge defends missing lecturer,' *South China Morning Post*, 1 June 1994; 'Battle for disputed funds proves futile,' *South China Morning Post*, 5 June 1994 and (letter) 'University was scrupulously fair,' *South China Morning Post*, 10 June 1994.