

Australian Recommendations on Computer Software Protection

Cristina Cifuentes
Department of Computer Science
University of Queensland
Brisbane Qld 4072, Australia
cristina@cs.uq.edu.au

Anne Fitzgerald
School of Law
University of Tasmania
Hobart Tas 7001, Australia
anne.fitzgerald@law.utas.edu.au

Abstract

We present the main recommendations of the Copyright Law Review Committee in its submission to the Australian Government in 1995 in relation to the protection of computer software in Australia and the intellectual property of software developers in this country.

The implications of these recommendations to the academic and industry computer science community are presented, as well as comparisons with European and US legislation.

1 Introduction

In Australia, computer programs have been protected by copyright as literary works under the Copyright Act [16] since 1984 when the Act was amended to specifically include software. However, the 1984 amendments [17] were regarded as a short-term measure and there was doubt about whether they were effective in ensuring that all computer programs were protected by copyright. As a result, the question of copyright protection of computer programs was referred to the Copyright Law Review Committee (herein referred to as the CLRC or the Committee) for inquiry in October 1988. The CLRC is a specialist advisory body which was first established in 1983 to inquire into and report to the government on specific copyright issues referred to it from time to time.

The Committee's terms of reference were

Whether the Copyright Act 1968, as amended by the Copyright Amendment Act 1984, adequately and appropriately protects computer programs in human and machine

readable forms, works created by or with the assistance of computer programs and works stored in computer memory.

The terms of reference were subsequently extended to include importation of computer programs (5 January 1989) and published edition copyright in relation to works stored in electronic databases (18 January 1991).

The CLRC released its *Draft Report on Computer Software Protection* (Draft Report) for public comment in June 1993 [18]. Following consideration of the submissions received in response to the Draft Report, the Committee published its *Final Report on Computer Software Protection* (Final Report) in April 1995 [19].

In the Final Report, the CLRC makes recommendations on an extensive range of issues, including the appropriate form of protection of computer programs, definitions of "computer program" and "reproduction", the exclusive rights of the copyright owner and the scope of exceptions to those rights, protection of computer-generated material and audiovisual works, and circumvention of program locks. After considering the comments on its draft recommendations, in the Final Report the CLRC was persuaded to change its recommendations on a number of important matters: the definition of reproduction, ownership of copyright, and parallel importation of computer programs. In the Final Report, the CLRC also revised its recommendations relating to computer-generated materials, exceptions to the copyright owner's exclusive rights, and the overlap between the Copyright Act 1968 and the Circuit Layouts Act 1989.

One aspect of the Final Report which has attracted considerable debate are the Committee's recommendations concerning exceptions to the copyright owner's exclusive rights which would, in specified situations,

permit reverse engineering involving decompilation. The introduction of limited decompilation rights is opposed by major hardware and software companies including IBM, Microsoft, Novell, Aspect and Computer Power. On the other hand, an alliance of companies known as the Supporters of Interoperable Systems in Australia (SISA), whose members include Fujitsu, Sun Microsystems, Amdahl and Storage Technology, support open systems and interoperability of software, and therefore are in favour of the proposed reforms.

This paper examines the main CLRC's recommendations on computer software protection. Section 2 looks at possible forms of protection of computer software, Section 3 explains the extent to which computer software is protected under copyright law, Section 4 reviews the exclusive rights of an author, Section 5 reviews the exceptions to the exclusive rights and Section 6 presents other recommendations made on matters such as databases, parallel importation and educational use of computer programs. Finally, Section 7 provides a discussion and conclusions on these recommendations.

2 Forms of Protection

Various forms of intellectual property protection are available in relation to computer programs, including copyright, patents and sui generis. In this section, each of these possible forms of protection is outlined.

Copyright protects the form of expression of ideas or information (but not the ideas or information in itself), conferring certain exclusive rights on the author or creator. Since the late 1970s, numerous jurisdictions around the world have enacted copyright protection for computer software, although this is not expressly required by the major international copyright treaty, the Berne Convention for the Protection of Literary and Artistic Works (1971) [4] (the Berne Convention). However, negotiations are presently nearing completion on a protocol to the Berne Convention which, among other changes, will make it clear that computer programs are to be protected as literary works for copyright purposes. This practice was recently confirmed in the Agreement on Trade-Related Aspects of Intellectual Property Rights (the TRIPs Agreement) which forms part of the General Agreement on Tariffs and Trade (GATT) of 1994. Article 10(1) of the TRIPs Agreement, which is binding on all members of the World Trade Organisation (WTO), provides that “[c]omputer programs, whether

in source or object code, shall be protected as literary works under the Berne Convention (1971).” Among the advantages of copyright protection are that it confers immediate protection from the time a program is created, without any need for registration formalities, and that it is recognised internationally under the terms of the Berne Convention.

Patent legislation in most countries provides for a patent to be granted for an invention which consists of patentable subject matter, is new, inventive, capable of industrial application and useful and which is disclosed in the patent specification. Until recently few patents were issued for computer programs and patent offices rejected applications for computer programs on the ground that they were not patentable subject matter. It has traditionally been accepted that patents cannot be granted in respect of naturally occurring phenomena, mental processes and abstract intellectual concepts¹. By analogy, mathematical algorithms incorporated in computer programs have been regarded as simply a method of calculating a mathematical problem or a scheme for operating a computer in a particular manner and, consequently, unpatentable subject matter. The objections to the patenting of computer software on these grounds have been gradually eroded by the courts in recent years. This has been particularly evident in the United States where the easing of the restrictions has led to a rapid escalation in the number of software patents being granted.

Sui generis protection has been advocated in recent years to create a new form of intellectual property protection for computer programs. This approach has been adopted where existing intellectual property systems cannot be adapted to fit a new technology. Perhaps the best known example of sui generis legislation is the United States Semiconductor Chip Protection Act 1984 and corresponding legislation enacted in other countries to provide protection for integrated

¹In the European Patent Convention (EPC) and legislation based upon it such as the UK Patents Act 1977, the exclusions from patentability are expressly stated. Article 52(2) of the EPC sets out a non-exclusive list of subject matters which are not to be regarded as inventions. These include:

- (a) discoveries, scientific theories and mathematical methods;
...
- (c) schemes, rules and methods for performing mental acts, playing games or doing business, and programs for computers; ...

circuits². A sui generis form of protection for software was proposed in 1977 by the World Intellectual Property Organisation (WIPO) which concluded that copyright protection was inappropriate for computer software and produced a set of *Model Provisions on the Protection of Computer Software* [28]. A more recent proposal for sui generis is that by Samuelson, Davis, Kapor and Reichman, *A Manifesto Concerning the Legal Protection of Computer Programs* [25], where the authors advocate for a new way of protecting software on the grounds that copyright is not suitable to protect program behaviour (and this is the most valuable aspect of a program) and many commercial program innovations are vulnerable to rapid, inexpensive copying by competitors. They propose a period of automatic anticloning protection for program innovations (1 to 5 years), so that there is greater incentive to invest in the development of innovative software. A proposal by Christie argues for the introduction of protection at the digital logic level and at the design level; not just at the source and object code levels [9, 10]. As can be seen, the advantage of sui generis protection is the greater flexibility in the protection of computer programs, as laws could be made with special regard to the nature of computer programs (both object and source code). However, this model failed to gain acceptance as countries increasingly turned to copyright protection.

Why Protect Computer Programs under Copyright Law?

On the fundamental question of the appropriate form of protection for computer software, the CLRC concluded that computer programs in either source code or object code should continue to be protected as literary works under the Copyright Act 1968. The Committee reached this conclusion with a perceptible degree of reluctance. In the Draft Report, the CLRC was of the view that computer programs would be optimally protected not as literary works under Part III of the Act, but as a new category of subject matter under Part IV which protects sound recordings, films, broadcasts and published editions. As attractive as this option may have been, the Committee recognised that to adopt it would place Australia in breach of its Berne Convention obligations. While the Committee did not regard a computer program in object code as

²For a comprehensive treatment of sui generis protection for integrated circuits, see A. Christie, *Integrated Circuits and Their Contents: International Protection*, Law Book Company, Sydney, 1995

being a literary work, it viewed a computer program in source code as a literary work which was required, under the Berne Convention, to be accorded the same protection as is given to traditional literary works.

However, in the Final Report, the CLRC saw any possibility of the introduction of the new sui generis copyright-style protection which it favoured as having been foreclosed by the TRIPs Agreement which was concluded in 1994. The Committee regarded the TRIPs Agreement as putting an end to arguments about the appropriate form of protection by emphatically requiring in Article 10(1) that all forms of computer programs be protected as literary works under the Berne Convention. To adopt the Committee's preferred option of introducing protection for computer software as a separate category in Part IV of the Act would place Australia out of step with its major trading partners and adversely affect its economic position. Australia is a net importer of computer hardware and software for use by government, educational institutions and commercial and industrial organisations and in the development of the domestic computer industry. Access to the latest computer technology would be jeopardised if the level of protection given to software under Australian law were perceived to be inadequate. Such considerations persuaded the Committee that computer programs should continue to be protected as literary works.

From an international perspective, the CLRC pragmatically saw Australia as having little choice but to continue to protect computer programs as literary works. The Committee recognized that there is now strong support world-wide for protection of computer programs as literary works, especially among our major trading partners including the United States, the United Kingdom and the European Community. Adoption of the CLRC's preferred solution preferred would put Australia out of step with developments in these jurisdictions and the growing international trend towards protection of computer software as literary works. Of greater concern to the Committee was the possibility that any unilateral action by Australia might be misinterpreted by countries "important to [our] economic and trading interests" as indicating a reduced commitment to the proper protection of computer programs. If such a perception were to lead to a reluctance on the part of overseas producers of software to export their latest products to Australia, there would be serious adverse effects on Australia's technological and economic development.

3 Copyright in Computer Programs

In 1984, the Copyright Act 1968 was amended to expressly provide protection for computer programs. The amendments were made hurriedly in reaction to the first instance decision of the Federal Court of Australia in *Apple Computer Inc v Computer Edge Pty Ltd* (1983) [1] where the Apple programs which had been reproduced in Computer Edge's Wombat brand computers were not considered literary works under the Copyright Act and hence were not protected by copyright. On appeal, the Full Court of the Federal Court upheld Apple's claim, holding that both the source code and object code were protected by copyright: *Apple Computer Inc v Computer Edge Pty Ltd* (1984) [2]. On further appeal, the High Court of Australia restored the trial judge's decision: *Computer Edge Pty Ltd v Apple Computer Inc* (1986) [14]. The High Court held that while the source code programs were original literary works for copyright purposes, the programs in object code were not literary works and could not be regarded as a reproduction or an adaptation of the source code and were therefore not protected by copyright. The consequence was that Apple was unable to prevent Computer Edge from importing and selling its Wombat computers which had copies of various Apple object code programs stored in ROM.

The Copyright Amendment Act 1984 amended the definitions in section 10(1) of the Act relating to "literary works" and "adaptation" and added new definitions for "computer program" and "material form". "Reproduction" was left undefined. These definitions were meant to make it clear that computer programs, whether in source or object code, are literary works within the meaning of the Copyright Act:

- "literary works" includes
 - (a) a table, or compilation, expressed in words, figures and symbols (whether or not in a visible form); and
 - (b) a computer program or compilation of a computer program.
- "adaptation"
 - (ba) in relation to a literary work being a computer program—a version of the work (whether or not in the language, code or notation in which the work was originally expressed) not being a reproduction of the work.

- "computer program"

computer program means an expression, in any language, code or notation, of a set of instructions (whether with or without related information) intended, either directly or after either or both of the following:

- (a) conversion to another language, code or notation;
- (b) reproduction in a different material form,

to cause a device having digital information processing capabilities to perform a particular function.

- "material form"

in relation to a work or an adaptation of a work, includes any form (whether visible or not) of storage from which the work or adaptation, or a substantial part of the work or adaptation, can be reproduced.

The definitions of "literary works" and "material form" contain the words "whether visible or not" to make express provision for object code programs which are stored in computer memory (harddisk, RAM, ROM) and are not "visible" (i.e. intelligible in the form of words, figures or symbols) to the human eye.

Since the 1984 amendments, there have only been two major Australian cases that have considered the effect of these provisions: *Autodesk Inc v Dyason* (1992) [3] and *Star Micronics Pty Ltd v Five Star Computers Inc* (1991) [27]. The CLRC saw this as an indication of the success of the present legislation. The purpose of the 1995 recommendations was to clarify uncertainties caused by comments by members of the High Court in the Apple case and to ensure a more comprehensive protection for computer programs.

Doubts had been expressed about whether the 1984 amendments actually achieved their intended purpose of conferring copyright protection on computer programs in object code as well as source code. These concerns were allayed by the High Court in *Autodesk Inc v Dyason*. The High Court rejected any narrow literal interpretation of the definition of "computer program" which would confine copyright protection to computer programs in some form of written expression. Such an approach would have frustrated the

obvious intent of the legislators which was to confer protection on the actual set of instructions regardless of whether they were expressed in a written form or stored in digital (non-sensate) form (e.g. on disk, ROM or EPROM). The High Court held that:

... the stored set of instructions in a non-sensate form such as electrical impulses is itself protected on the basis that copyright actually subsists in any expression or description of it which can theoretically be made in language, code or notation.

The High Court's decision in *Autodesk Inc v Dyason* has dispelled any doubts about the effect of the 1984 amendments. It is now clear that the definition of "computer program" in s.10(1) applies to computer programs in both source code and object code.

3.1 Changes to the 1984 Definitions

After considering the submissions made in relation to the definition of "literary works", the Committee recommended that the words "whether or not in a visible form" be deleted from the definition of "literary work". Similar words form part of the definition of "material form", and since the definition of "literary works" includes a special entry for computer programs, then it is not necessary the amendment on entries not explicitly relating to computer programs.

Submissions were received to delete the definition of "computer program" from the Act. In fact some countries of the European Union do not provide for such a definition in its legislation. The Committee was of the opinion that a definition is needed to provide certainty and clarity in the law at this stage, but that with the increasing familiarity of technology and concepts in computer programs, a definition may not be required in the future. Both the US and Japan provide for definitions in their legislation. The Committee recommended that the definition of "computer program" be substituted by the US definition stated in section 101 of the US Copyright Act 1976

A "computer program" is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result.

This definition provides several advantages over the present one as it is not limited to programs for digital

computers, it clearly extends to include programs written in declarative programming languages and it covers programs in source code, object code and microcode.

The Committee declined to provide a definition of "computer", as had been urged in a number of submissions. Its preference was for the meaning of computer to be left to the courts to determine on a case by case basis, applying a normal understanding of what a computer is.

Finally, the Committee noted that the definition of "material form" is an inclusive, non exhaustive, definition which should be construed as including forms of storage where a work or an adaptation of a work exists in a form which would not normally be regarded as material, such as electronic and magnetic forms of storage, but which are amenable to reproduction.

3.2 Ownership

Section 35(2) of the Act provides the author of a literary work with the ownership of any copyright subsisting in the work. Joint authorship is also granted when programs are written by a group of people who collaborated towards the same program. However, there are circumstances when programs are written by different people who are not collaborating jointly towards the same program but who write independent modules. In such case, individual copyright is granted to individual authors of independent modules.

Section 35(6) provides that where literary works are made by authors in pursuance of the terms of their employment under a contract of service or apprenticeship, then the employee is the owner of any copyright subsisting in the work. It is also noted that both parties can negotiate for different ownership of copyright.

The Australian Vice Chancellor Committee and the Australian Committee of Directors and Principals Limited, in their submission to the CLRC [19], stated that there is an issue regarding ownership of programs written by staff and/or students in educational institutions, particularly when these programs were produced by the use of equipment and other resources (e.g. compilers) in the institution. Reference was made to section 35(6) of the Act, which provides, in relation to a staff member, that ownership of copyright would depend upon whether or not the work was created in pursuance of the terms of his/her employment. However, in the case of students, ownership resides

with the student as there is no contractual employment between students and educational institutions. The CLRC felt that no recommendation was required regarding this issue.

3.3 Duration

The duration of copyright of literary and artistic works is governed by Article 7 of the Berne Convention [4], which states that protection is granted for the life of the author plus 50 years.

In Australia, section 33(1) of the Act grants different duration of copyright protection depending on whether the work was published or not. In the case of published work, the duration extends for 50 years after death of the author, but in the case of unpublished work, copyright is indefinite.

The CLRC discussed that there should not be a distinction made between published and unpublished computer programs given the utilitarian purpose of programs and the useful life time of them. Given the recent TRIPs agreement, the Committee felt obliged to recommend that

- (a) The existing term of protection consisting of life of the author plus 50 years should apply to all computer programs, whether published or unpublished.

There are several disadvantages with the terms of duration of copyright of computer programs as it is not only too long (given the technological advances in any 50-year period), but it is unreasonable to protect different computer programs – especially competing programs – with different periods of time based on the length of the lives of their respective authors. This information will also be hard to track down in the case of joint authorship of a program, where the 50-year period of protection starts with the death of the last author.

4 Exclusive Rights

The owner of copyright in a computer program has the same exclusive rights as are provided for in section 31 of the Copyright Act in relation to literary works. In the case of a literary work, s.31(1) of the Copyright Act provides that the owner of copyright has the exclusive right to:

- (a) reproduce the work;
- (b) publish the work;
- (c) perform the work in public;
- (d) broadcast the work;
- (e) cause the work to be transmitted to subscribers to a diffusion service;
- (f) make an adaptation of the work; and
- (g) do any of the acts (a) to (e) in relation to an adaptation of the work.

to the extent that these rights are applicable to computer software.

In other words, the owner of copyright in a computer program will enjoy the full bundle of rights set out in s.31, including the right to control the reproduction of the computer program and the right to make an adaptation of the work. Although the exclusive rights of the copyright owner vary in different countries (e.g. US, UK and Germany), almost all world countries have agreed to the Berne Convention and the GATT TRIPs Agreement, hence they are required to recognise the rights set out in these agreements (i.e. protect computer programs as literary works) in their own copyright legislation.

Right (g) limits the owner's rights on adaptations of the copyrighted work, as once an adaptation has been made, the owner has no control over that adaptation. In the case of computer programs, the important issue is therefore what is considered a reproduction of a program and what is an adaptation of a program. This uncertainty has been created for the lack of a definition of reproduction in the Act. The courts have interpreted reproduction to mean copying.

4.1 Object Code vs Source Code – Adaptation or Reproduction?

Applying the existing definition of “adaptation” (see Section 3), the Committee was of the view that an object code version of a program is in fact an *adaptation*, rather than a *reproduction*, of the source code program. However, the computing community regards object code as a reproduction of source code via the compilation process, as the object code program is the machine representation of the source code program. In order to cater for this difference between the law and the general understanding of object code in the

computing community, an explanatory provision in regards to “reproduction” should be introduced to provide that, for a computer program, a “reproduction” includes, but is not limited to:

- (a) an object code version of the program that has been derived from the program in source code by compilation; and
- (b) a source code version of the program that has been derived from the program in object code by decompilation.

The Committee also clarified that – in relation to works stored electronically – the conversion of a work (or an adaptation of the work) from its hard copy human readable form to an electronic form of storage is a reproduction of the work (or the adaptation). Similarly the instance of converting a work (or an adaptation) from an electronic form to a hard copy, such as making a print out of a work stored electronically should also be considered a reproduction of the work.

If the clarifications of reproduction are introduced in the Copyright Act, the scope of the definition of adaptation would subsequently be narrowed. As a result, adaptation would be limited to translations of programs that do not include compilations or decompilations, therefore meaning translations of programs at the same level of abstraction (i.e. high-level to high-level translations or low-level to low-level translations). Throughout the Final Report the discussions made it clear that adaptation was effectively covering only different source code versions of programs written in source code (i.e. high-level translations). However, with the emergence of binary translation technology [26], i.e. the conversion of an object code program for one platform to an object code program for a different platform, these low-level translations will also be covered by the adaptation definition.

4.2 Protection of Non-literal Elements of a Program

It is difficult to ascertain the appropriate scope of the reproduction right because of the unique nature of computer programs as functional copyright works which cause computers to perform certain functions. Some aspects of computer programs which have commercial value and, arguably, deserve protection fall outside the proper scope of copyright protection. The Committee considered whether copyright protection should

extend to various aspects of computer programs which owners had sought to protect such as “look and feel”, “user interface”, “non-literal elements” and “structure, sequence and organisation”. The Committee considered that for most users of computer programs, the distinguishing feature of the program is its “look and feel” or “user interface”.

It is widely known that a program is composed of *code* (i.e. text) which causes it to *behave* in a certain way once it has been compiled. These two elements are referred to as the *literal* (i.e. textual) and *behavioural* (i.e. functional) elements of a computer program. Both types of elements are said to have their own “structure, sequence and organisation”. For program code, the “structure, sequence and organisation” roughly equates to the organisation and layout of chapters, sections, index and so on in a technical literary work. For program behaviour, the “structure, sequence and organisation” are the features which give it user appeal, such as screen icons, screen displays and the sequence of screen displays. A program’s “non-literal elements” include both the structure, sequence and organisation of a program’s behaviour and the structure, sequence and organisation of its underlying code.

The non-literal elements of a program are protected in the same way as the structure, sequence and organisation of traditional literary works. The Committee supported the approach developed in the US case *Computer Associates International v Altai Inc* for determining whether the non-literal elements of program code had been infringed. It considered the three-part abstraction-filtration-comparison test set out by the Altai court to be “a very practical and useful guide for determining infringement of computer programs”.

The desirability of promoting standardisation of user interfaces and ensuring that the most efficient user interfaces are used and developed were seen to outweigh the need to grant authors copyright protection for the “look and feel” of program behaviour. Given that screen displays generated by a computer program can be regarded as an aspect of the program’s behaviour, the Committee recommended against the introduction of any additional protection for screen displays.

5 Exceptions to Exclusive Rights

Exceptions to the exclusive copyright owner rights are introduced in legislation in order to allow for the

normal and fair use of computer programs in the community. These include intermediate reproduction of the programs in the computer's random access memory (RAM) when the program is ran, the making of back-up copies, and the reverse engineering of the program to determine interfaces for interoperability and error correction.

5.1 Copying for Normal Use and Back-up Copying

When a program is used, an intermediate copy (i.e. a reproduction) of the program is made in the computer's RAM in order to run the program. This intermediate reproduction amounts to an exercise of the copyright owner's exclusive right to reproduce the program. However, this intermediate copy is needed for the normal use of the program, and therefore an exception to permit copying of programs for normal use was recommended. However, the Committee did not feel the need to define "normal use" of a program.

In a similar case of normal use of computer programs, the computer community is used to making a copy of a program for back-up purposes. This reproduction of the program also amounts to an exercise of the copyright owner's rights, and therefore an amendment to s.43A of the Act was recommended to permit the owner of a copy of a computer program to make an ephemeral back-up copy of a program which may be stored and used in place of the original if the original is lost, destroyed or becomes unusable. Where the original or back-up copy is destroyed or damaged, the surviving original or back-up may be used to make another back-up copy. However, the right to make a back-up copy would not extend to a program which has been "locked"³ by the copyright owner against copying.

5.2 Reverse Engineering and Decompilation

The Committee was concerned that permitting reverse engineering of programs by decompilation would make it relatively easy for competitors to produce clone programs, thereby obtaining a free ride on the efforts of the creator of the original program. As a consequence, the Committee recommended that reverse engineering involving decompilation should be prohibited except

³A "locked" program is one that has software or hardware protection to prevent the copying of the program.

to the extent that it is required for interoperability or error correction.

The definitions used in the Final Report for decompilation, disassembly and reverse engineering were [19]:

- **Decompilation**
The working back from the object code of a computer program to a version of the source code. This process may involve a substantial recreation or reproduction of the source code of the original program. Decompilation is achieved using a computer program called a decompiler.
- **Disassembly**
The working back from object code to assembler code ie, a special case of decompilation. Disassembly is achieved using a computer program called a disassembler.
- **Reverse engineering**
The study or analysis of a computer product (including a computer program) in order to reveal the underlying idea or principle on which it operates. This analysis may include an examination of relevant published documentation, study of the operation of the product and, in the case of a computer programs [sic], their decompilation. Studying the operation of a program would involve reproduction of the program in the same way as normal use.

However, throughout the Final Report's discussions, the terms decompilation and disassembly were treated as synonymous. This causes an overgeneralization of some of the techniques used in these areas.

A presentation by IBM for the benefit of the Committee demonstrated the disassembly of a relatively small program [19]. The Committee accepted IBM's assertion that there are computer programs available which can very rapidly decompile other programs, thus making it possible to produce clone programs relatively easily. IBM also asserted that once a program is decompiled into a high level language, a competitor wanting to produce a clone can easily manipulate the code derived from decompilation so as to hide any visual similarity to the original program, while retaining the same functionality.

However, 5 years after IBM's demonstration, there are still no commercial decompilers available on the market, and the few disassemblers that are available have severe limitations. Reverse engineering techniques to

detranslate object code programs are not straight forward, especially if the level of abstraction that the user wants to arrive to is that of a high-level language. Reverse engineering at the disassembly level has been used to create clone programs of microcode and operating system code (i.e. relatively small programs when compared to programs generated from high-level languages). Reverse engineering at the decompilation level requires a lot of work, time, and effort to get it right. The automation of this process is incomputable in general due to its equivalence to the halting problem [23], which means that fully automated static disassemblers and decompilers will never be available unless additional information, which is not normally required by the machine (computer), is stored with the object code [13]. Also, today's programs are very complex and include a variety of services provided by the operating system, for example, graphical routines, which make it hard to very rapidly decompile, modify and produce a clone program.

The exceptions dealing with decompilation are explained in the next two subsections. These exceptions are grouped by those exceptions that are controversial (and were also controversial in Europe when recommended and then introduced in the legislation [8, 21, 22, 5]) and those that are generally accepted as fair use.

5.3 Interoperability and Error Correction

The Committee recommended that decompilation 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 [19]

- (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 authorised to do so;
- (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.

In other words, decompilation for the purposes of determining the interface to another program or hardware device is permitted, so long as the specifications of that interface have not been made available, and

that the decompilation process is only confined to the parts of the program necessary to retrieve the required interface. This exception is inline with the growing support for interoperable and open systems: if the developer does not make the interface available, then the interface can be determined by means of disassembly or decompilation.

In the Final Report, the CLRC revised its draft recommendations on decompilation for error correction, proposing amendment of the Copyright Act to provide that decompilation of a program for error correction does not infringe copyright where an error free version cannot be obtained within a reasonable time at a normal commercial price. What amounts to a "reasonable time" or a "normal commercial price" will be determined in individual cases.

In other words, if a program has a bug and the copyright owner of that program cannot provide a bug-free version of the program at a normal commercial price and within a reasonable time, then it can be decompiled for patching purposes. Although a software vendor or user would only resort to disassembly or decompilation to fix an error in extreme circumstances (e.g. when the software developer is out-of-business), it is unclear why a user would have to pay a normal commercial price for the bug-free version of the software once that software has been bought once. It is reasonable to expect to pay for recovery of material costs (e.g. floppies and CDs) and postage, but not to pay twice for the same software product.

Both of these exceptions make it clear that decompilation should be limited to those acts necessary to achieve interoperability or to correct the error, as the case may be. However, given the nature of object code, it will often be difficult to determine which particular pieces of the code are relevant to the interoperability issue or contain the error. This means that the acts of decompilation or disassembly may have to be performed on the entire object code program before actually determining what piece of code is required to be analysed – how could you prove in court that this really was required based on these recommendations and the understanding of computer programs in object code?

The CLRC's recommendation on decompilation for interoperability closely resembles Article 6 of the EC

Directive on the Legal Protection of Computer Programs. Amendment of the Copyright Act in accordance with the CLRC's recommendation regarding decompilation for interoperability is presently being opposed by a number of large computer software and hardware companies. On the other hand, it is argued that the CLRC's recommendations are too narrow and that by limiting the permissible scope of decompilation to the cases of interoperability and error correction, the recommendations would confer copyright protection on *functional* aspects of computer programs which should not be protected. If implemented in legislation, the limited decompilation right proposed by the CLRC would arguably put Australian programmers and software companies at a disadvantage when compared with their counterparts in the United States, where a broader right to decompile has been recognised by the courts [15]. For a complete discussion on the differences between US and Australian copyright law refer to [12].

5.4 Decompilation to Understand Techniques and Defeat Program Locks

The Committee recommended that decompilation for the purposes of understanding techniques is governed by the fair dealing⁴ provisions of the Copyright Act adding the qualifier that the fair dealing provision only applies to "non-commercial" activities.

The use of the term "non-commercial" in this context is likely to cause uncertainty. Decompilation to understand ideas or techniques by university researchers may be regarded as non-commercial study or research which amounts to a fair dealing under s.40. However, the usefulness of the commercial vs non-commercial distinction becomes increasingly questionable and difficult to apply as universities enter into research joint ventures or license their research results to commercial organizations.

To cover "black box" reverse engineering which does not involve decompilation of object code, the Committee recommended that the Copyright Act should be amended to allow the reproduction and study of computer programs in the circumstances contemplated in Article 5(3) of the EC Directive on Legal Protection of

⁴Section 40(1) of the Copyright Act provides that:

A fair dealing with a literary ... work, or with an adaptation of a work, for the purposes of research or study does not constitute an infringement of the copyright in the work.

Computer Programs. The Committee regarded such a provision as necessary in the light of the High Court's decision in *Autodesk Inc v Dyason* which held that copyright in the plaintiff's program was infringed by black box reverse engineering.

The Committee also recommended that the modification of a locked program to circumvent the lock, and subsequent copying of the program, should be prohibited unless done with the copyright owner's consent. The terms of this recommendation are similar to Article 7 of the EC Directive on the protection of computer software and s.296 of the UK Copyright, Designs and Patents Act 1988. Notwithstanding this recommendation, the Committee emphasized that users of computer programs will still be able to circumvent locking devices for legitimate purposes, e.g., by decompilation for error correction or to create an interoperable program. Users of computer software would bear the onus of understanding the circumstances in which circumvention of a program locking device is permissible.

5.5 Matters Left for Negotiation

Several recommendations leave matters for negotiation between the copyright owner and the relevant party (the purchaser of the program in most cases); these include:

- the copying of programs by third party maintainers;
- the right to modify computer programs for enhanced performance;
- the right to modify programs to run on networks; and
- the right to decompile for porting purposes.

The CLRC did not make any comments on what procedures should be available to a user when the copyright owner of a program does not want or is not able to create an enhanced version of or port the program (either due to time or financial constraints). There is also the possibility that the copyright owner has gone out of business, in which case, there may not be anyone for the user to negotiate with.

With the advent of newer and faster machines, the computer industry is in need of software tools that aid in the porting of applications from one computer to another (i.e. binary translation). These tools

will require analysis of object code programs in order to translate them to another computer configuration. This translation can be done in two different ways: statically or dynamically. In the case of static translation, the binary translator program creates a new object code program for the new machine. This new file would constitute an adaptation of the original object code program and would fall within the copyright owner's exclusive rights. In the case of dynamic translation, the binary translator performs the translation "on the run" by interpretation of the object code program on the new machine. No new object code program is created, although the object code is reproduced in RAM in running the program, which is considered an exception to exclusive rights for the normal use of computer programs. It is not clear whether running the program for this purpose can be regarded as copying which is reasonable or necessary for the normal use of the program and thus within the scope of the CLRC's recommended exception.

6 Other Issues

6.1 Works Stored in Computer Memory: Databases

In regards to electronic databases, the Committee drew attention to Article 10(2) of the TRIPs agreement and concluded that, for the purposes of copyright law, there was no reason to apply different principles to electronically stored databases and those stored in other forms. As a result, the Committee's conclusions do not distinguish between traditional hard-copy databases, databases stored in computer memory, and databases stored in other forms of storage such as floppy discs, tapes and CD ROMs.

The Committee considered the question of the duration of copyright protection for "dynamic databases", i.e. electronic databases which are being constantly updated. It rejected suggestions that copyright protection for dynamic databases is potentially eternal, pointing out that it is possible to identify a time when the first edition of any published database is made. While insubstantial additions, such as merely adding information, do not give rise to a new edition, the eventual cumulative effect of constant updating will amount to a sufficiently substantial change to the database such that it may be regarded as a new edition in which copyright subsists. In determining whether a new edition has been created, matters such as the

quality and quantity of the material added and the way it has been incorporated into the database will be taken into account. As the question of when first and subsequent editions of computerised databases come into being are evidentiary issues to be determined according to the circumstances in individual cases, no legislative amendment was required in relation to duration of copyright protection.

To the Committee, the principal distinguishing feature of a computer database is the fact that it can be connected to a network of terminals from each of which it can be accessed and viewed. The capacity to search and view works included in an electronic database increases the value of those works to both the copyright owner and the database operator, which may be reflected in the royalty negotiated by the copyright owner when licensing the inclusion of the work in the database.

The recommendations on protection of databases only encompass traditional copyright protection and are minimal when compared to the recommended European EC Database Directive (April 1996) which introduces dual forms of protection for databases: copyright and sui generis. The former uses the traditional criteria for copyright protection; the latter is a new form of protection for databases that require time and money to put together but which would otherwise not attract copyright protection.

Screen Displays

The viewing of a database on a screen display raised the question of whether a screen display is a reproduction in a material form of the work or other material displayed. In the Draft Report, the CLRC was of the view that as screen displays are ephemeral they cannot be "a form of storage". In the Final Report, the Committee acknowledged that there was, however, some doubt as to whether screen displays involve a reproduction in a material form of works stored in computer memory, although the better view from a policy perspective was that they did not. The Committee affirmed its draft recommendation that the Copyright Act should be amended to make it clear that screen displays do not constitute a reproduction in a material form of works stored in computer memory. This recommendation recognises that "electronic browsing" is a normal use of a computer database, just as the reader of a book is able to browse the pages of the book without exercising any of the copyright owner's rights.

The Committee also examined the question of whether the screen display of materials stored in a database could be regarded as a public performance. The Committee affirmed that its intention was to recommend that the mere act of calling up a work from a computer database onto a computer terminal should not, by itself, constitute a public performance of the work and the Act should be amended to reflect this. Whether a screen display is in fact a public performance will depend upon whether it is carried out "in public". The Committee recognised that screen display is likely to become an increasingly frequent means by which copyright works will be used. However, it did not regard the right of public performance as a suitable vehicle for controlling all acts of display on screen without distorting the notion of public performance as it is presently understood in the Copyright Act.

Liability of Database Operators

In regards to the liability of database operators, it was recommended that a provision be included in the Copyright Act to the effect that where the licence of a database provider does not extend to authorising the making of a hardcopy/ies by a database user, the networking of the database to subscribers would not, of itself, amount to authorisation of the making of such copies if the operator arranges for a message to advise subscribers of their copyright liability for such reproduction.

Electronic Copies made by Libraries

In relation to copies made by libraries and educational institutions, amendments to sections 49 and 50 are required to ensure that libraries are able to make electronic copies (including electronic transmission of a copy and the loan of an electronic copy) available to library users within the limits prescribed by these sections and subject to payment of royalties where applicable.

To this end, the ACM has now changed its copyright notice to explicitly allow for electronic or digital copies [6]:

Copyright (C) 199x by the Association for Computing Machinery, Inc. Permission to make **digital** or hard copies of part or all of this work for personal or classroom use is granted [...] [emphasis added]

6.2 Educational Use of Computer Programs

It was recommended, in accordance with the expressed preference of most educational institutions and makers of computer programs, that a statutory licence scheme for the educational use of computer programs should not be established, but that the Government should review this situation in three years time.

6.3 Computer-generated Materials

In the Final Report the Committee differentiates between materials created with the assistance of computer programs and materials generated by computer programs where a human author cannot be identified. In the former case, a human author uses the computer programmed with software as a tool to facilitate the production of new materials, e.g., where a literary work is written using a word processing program. Materials produced with the assistance of computer software will attract copyright protection in the same way as those produced by traditional means, provided the requirements of the Copyright Act are satisfied, and the usual rules regarding ownership and duration of copyright will apply.

The latter case identified by the CLRC is where material is generated by a specially programmed computer but it is not possible to identify any human author. The CLRC agreed that in the absence of a human author, computer-generated materials could not be properly classified as "works" which pre-suppose human authorship and originality. The Committee considered that this type of computer-generated material was more appropriately protected as subject matter other than works, and hence it recommended that a new category of subject matter; "computer-generated material"; should be added to the Act and that a definition of "computer-generated" similar to that in the UK Copyright, Designs and Patents Act 1988 should be adopted, providing that

"computer generated", in relation to computer-generated material means that the material is generated by computer in circumstances such that there is no human author of the material.

To overcome the problem of attributing authorship of computer-generated material, the Committee recommended that the Copyright Act be amended along the

lines of the UK legislation to provide that the author of computer-generated material is the “person by whom the arrangements necessary for the creation of the material are undertaken”. Computer-generated material should be protected for a term of 25 years from the end of the year in which it was made.

7 Discussion and Conclusions

During recent years there has been much debate about whether computer programs should be protected by copyright, patent, or some sui generis or hybrid category. It has been argued that the most valuable feature of a computer program is its behaviour, which is not protected by the existing copyright or patent paradigms. Hence, there have been calls for the development of a new sui generis regime for the protection of computer software which would operate alongside the copyright and patent systems [25, 9, 10]. While such proposals have attracted some interest, it is unlikely that they will be given effect in the near future. The challenge for courts and legislatures at present is how best to accommodate computer software within the existing copyright and patent frameworks.

In the Final Report, the CLRC had the opportunity of examining the respective roles of sui generis, copyright and patent protection. It regarded the possibility of sui generis protection as being foreclosed by Article 10(1) of the GATT TRIPs Agreement and minimised the role of the patent system in protecting computer software. The Committee found little support for patent protection in the submissions received, concluding that it did not “regard as adequate, or support, the development of patent or patent-type protection.” It noted in passing that patent protection is available for computer software “in some cases”, referring to the Federal Court decisions in *IBM Corporation v Commissioner of Patents* (1991) [24] and *CCOM v Jiejing Pty Ltd* (1994) [7]. In fact, these decisions make it clear that most forms of computer software inventions are now patentable in Australia provided they satisfy the other statutory criteria.

The CLRC’s limited discussion of the role of patents appears to be based on its view of patenting as an alternative, rather than an adjunct, to copyright protection. By too readily dismissing the relevance of patent protection, the CLRC missed the opportunity of carrying out a detailed examination of the respective roles

of the copyright and patent systems in relation to computer software. Copyright law is concerned with protecting the expression of ideas and plays a limited role in protecting function. The ideas and techniques underlying a computer program may, if sufficiently novel and inventive, be protected by patents but are not protected by copyright.

The 1984 amendments to the Copyright Act have been given a wide interpretation in the recent decisions in *Autodesk Inc v Dyason* (1992) [3] and *Data Access Corporation v Powerflex Services Pty Limited* (1996) [20]. In these cases, the courts have ruled that anything that is essential to the operation of the program is a substantial part of the program and therefore will infringe copyright if reproduced, and that copyright can be infringed even if the copy of the program is not itself a program (e.g. a table).

These decisions have set a very low threshold for copyright infringement of computer programs and leave very little scope for reverse engineering or creation of interoperable software or hardware. The CLRC’s recommendations concerning the creation of a specific exemption to permit decompilation for interoperability would, if implemented, go some way towards addressing this problem. However, the Committee’s view that reverse engineering for the purpose of understanding the techniques underlying a computer program would amount to an infringement conflicts with the basic principle that copyright protects the expression of ideas rather than the ideas per se. The techniques underlying a computer program are its ideas rather than its expression, and copying them should not infringe copyright in the program itself [11].

The issue of the appropriate scope of patent and copyright protection for computer software is something which will have to be addressed by the government in its response to the CLRC’s Final Report. At present, the extensive scope of copyright protection applying to computer software in Australia means that most software developers rely primarily on copyright. Unlike the situation in the United States, the recent decisions which have established a wide scope for the patenting of computer software have not resulted in a marked increase in patenting activity.

The computing community needs to be more aware and proactive towards legislation on computer software protection, as such legislation affects not only software developers but also researchers in academic institutions; particularly when interested in commercialization of research results.

Acknowledgments

We would like to thank the anonymous referees for their insights and comments on this issue.

References

- [1] Apple Computer Inc v Computer Edge Pty Ltd. 50 ALR 581, 1983.
- [2] Apple Computer Inc v Computer Edge Pty Ltd. 53 ALR 225, 1984.
- [3] Autodesk Inc v Dyason (No 1). 22 IPR 163, 1992.
- [4] Berne Convention for the protection of literary and artistic works, Paris Act, 1971.
- [5] G. Burkill. Reverse compilation of computer programs and its permissibility under the Berne Convention. *Computer Law & Practice*, 6(4):114–119, March 1990.
- [6] ACM interim copyright policies. *Communications of the ACM*, 38(4):104–107, April 1995.
- [7] CCOM v Jiejing Pty Ltd. 28 IPR 481, 1994.
- [8] M. Cheek. Decompilation questions slow implementation of EC software directive. *IEEE Software*, 10(2):97, March 1993.
- [9] A. Christie. Re-writing the rules on the form of protection for computer software. *Journal of Law and Information Science*, 4(2):224–241, 1993.
- [10] A. Christie. Designing appropriate protection for computer programs. *European Intellectual Property Review*, 11:486–493, 1994.
- [11] A. Christie. The future of Australian copyright protection for computer programs and related products. *Journal of Law and Information Science*, 7(1):87–104, 1996.
- [12] C. Cifuentes and A. Fitzgerald. Reverse engineering of computer programs: Comments on the copyright law review committee’s final report on computer software protection. *Journal of Law and Information Science*, 6(2):241–276, December 1995.
- [13] C. Cifuentes and K.J. Gough. Decompilation of binary programs. *Software – Practice and Experience*, 25(7):811–829, July 1995.
- [14] Computer Edge Pty Ltd v Apple Computer Inc. 161 CLR 171, 1986.
- [15] P. Conrick. Nettle slips through CLRC’s gloves. *Australian Intellectual Property Law Bulletin*, (8):49–51, 1995.
- [16] Copyright Act 1968 (Cth).
- [17] Copyright Amendment Act 1984 (Cth).
- [18] Copyright Law Review Committee. *Draft Report on Computer Software Protection*. Office of Legal Information and Publishing, Australian Capital Territory, June 1993.
- [19] Copyright Law Review Committee. *Computer Software Protection*. Copyright Law Review Committee, Office of Legal Information and Publishing, Australian Capital Territory, April 1995.
- [20] Data Access Corporation v Powerflex Services Pty Limited. 33 IPR 194, 1996.
- [21] ECIS. The copyright (computer programs) regulations 1992: comments from ECIS. *Computer Law & Practice*, 8(5):156–157, 1992.
- [22] FAST, BSA, and SPA. The copyright (computer programs) regulations 1992: comments from FAST, BSA and SPA. *Computer Law & Practice*, 8(5):157–161, 1992.
- [23] R.N. Horspool and N. Marovac. An approach to the problem of detranslation of computer programs. *The Computer Journal*, 23(3):223–229, 1979.
- [24] IBM Corporation v Commissioner of Patents. 22 IPR 417, 1991.
- [25] P. Samuelson. Software compatibility and the law. *Communications of the ACM*, 38(8):15–22, August 1995.
- [26] R.L. Sites, A. Chernoff, M.B. Kirk, M.P. Marks, and S.G. Robinson. Binary translation. *Communications of the ACM*, 36(2):69–81, February 1993.
- [27] Star Micronics Pty Ltd v Five Star Computers Inc. 22 IPR 473, 1991.
- [28] World Intellectual Property Organisation. Model provisions on the protection of computer software, 1977.