COM6655 Professional Issues
Autumn 2021-22

Intellectual Property (part 2)

Professor Guy Brown
Department of Computer Science, University of Sheffield
g.j.brown@sheffield.ac.uk

Aims of this lecture

- To explain how the Copyright, Designs and Patents Act (1988) applies to computer software
- To explain exceptions relating to the Copyright (Computer Programs) Regulations 1992
- To review problems with the CDPA
- To discuss the 'look and feel' as a basis for copyright infringement

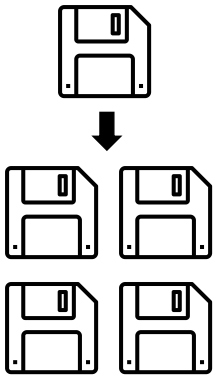
The CDPA and computer programs

Copyright in computer programs

- The CDPA states that for copyright to subsist in a computer program it must be original and recorded.
- **Original**: the program should be the result of a modest amount of skill, labour or effort and that it originates from the author.
  - Practically all computer programs meet the requirement of originality.
- **Recorded**: using "any form of notation or code, whether by hand or otherwise and regardless of the method by which, or medium in or on which, it is recorded.'

Restricted acts for computer programs

- Restricted acts for computer programs are:
  - copying
  - issuing copies to the public, including rental
  - making an adaptation
- Restricted acts are things you cannot do without the permission of the copyright holder.
- But there are some exceptions … see later.



Copying

- Copying means reproducing the work in any material form, including electronic storage.
- Copying also includes the making of copies that are **transient or incidental** to some other use of the work.
- Hence, the act of loading a program into a computer in order to execute it is considered 'copying'.
- So, an unauthorised **use** of a computer program will infringe copyright in that program.
- This is why a **licence** is required to use software.

What is a 'substantial part' of a program?

- Copyright protection would be useless if infringement could only occur by copying the entire work.
- **Q. Why?**
- The CDPA states that the exclusive rights apply to the whole or a substantial part of the work.
- **Q. How should 'substantial part' be defined?**

Quality rather than quantity

- Case law has established the following in relation to a 'substantial part' of a computer program.
- The test of infringement is one of **quality** not **quantity**.
- Copying even a very small part of a program will be an infringement if that part is **significant in terms of the program's function**.

IBCOS Computers Ltd. v Barclays Mercantile Highland Finance Ltd (UK) (1994)

- We know that computer programs are subject to copyright. But what about software collections?
  - Programmer at IBCOS worked on an accounts package
  - Later, produced a competing package for Barclays
- Copyright was held to subsist:
  - in programs **and** in the way programs are structured/linked together
  - depends on how much skill and judgment is involved in creating them

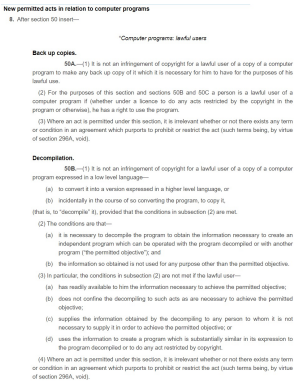
Making an adaptation

- Making an adaptation of a work includes **translating** it.
  - "In relation to a computer program a 'translation' includes a version of the program in which it is converted into or out of a computer language or code or into a different computer language or code, otherwise than incidentally in the course of running the program."
- This provision aims to control the **decompilation** and **disassembly** of computer programs.
- However … the Copyright (Computer Programs) Regulations 1992 permit decompilation under certain circumstances.

Decompilation and disassembly

Exceptions to copyright infringement

- The Copyright (Computer Programs) Regulations 1992 were prompted by an EC directive on the legal protection of computer programs. They grant
  - the right (sometimes) to decompile
  - the right to make back-up copies
  - the right to copy or adapt for purposes of error correction.
- Also, the CCPR introduce a defence of public interest, e.g. if it is in the interest of the public that a program listing should be published.



New permitted acts in relation to computer programs

8. After section 50 insert—

'Computer programs: lawful users

Back up copies.

50A.—(1) It is not an infringement of copyright for a lawful user of a copy of a computer program to make any back up copy of it which it is necessary for him to have for the purposes of his lawful use.

(2) For the purposes of this section and sections 50B and 50C a person is a lawful user of a computer program if (whether under a licence to do any acts restricted by the copyright in the program or otherwise), he has a right to use the program.

(3) Where an act is permitted under this section, it is irrelevant whether or not there exists any term or condition in an agreement which purports to prohibit or restrict the act (such terms being, by virtue of section 296A, void).

Decompilation.

50B.—(1) It is not an infringement of copyright for a lawful user of a copy of a computer program expressed in a low level language—

(a) to convert it into a version expressed in a higher level language, or

(b) incidentally in the course of so converting the program, to copy it,

(that is, to "decompile" it), provided that the conditions in subsection (2) are met.

(2) The conditions are that—

(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.

(3) In particular, the conditions in subsection (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.

(4) Where an act is permitted under this section, it is irrelevant whether or not there exists any term or condition in an agreement which purports to prohibit or restrict the act (such terms being, by virtue of section 296A, void).

Decompilation of computer programs

- Making an adaptation includes decompilation and infringes copyright unless allowed by the new decompilation right.
- By the new right, a **lawful user** of a program may decompile a program if *necessary to obtain the information necessary to achieve the interoperability of an independently created program with the decompiled program*.
- But the decompilation right does not apply if the developer intends to use information obtained by decompilation to develop a **competing** product.

Backup copies of computer programs

- The CCPR states that copyright is not infringed by a lawful user making a copy of a program for backup purposes if doing so is necessary to lawful use.
- "*it is irrelevant whether or not there exists any term or condition in an agreement which purports to prohibit or restrict the act*"

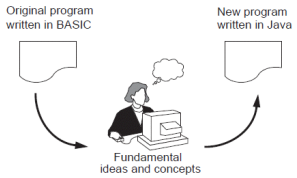
Error correction and other exceptions

- The CCPR permits a lawful user to adapt a program provided:
  - it is necessary to do so for lawful use, and
  - adaptation is not prohibited by any agreement regulating the user's lawful use
- "*It may, in particular, be necessary for the lawful use of a computer program to copy it or adapt it for the purpose of correcting errors in it.*"
- **Q. When is it necessary to copy or adapt software to correct errors? When is it necessary to make a backup for lawful use?**

Problems with the CDPA

Is the copyright law too strong?

- The meaning of 'translation' in the CDPA may be too wide, since it appears to include the **manual** conversion of a program from one language to another.
- However, rewriting a program requires skill and effort to identify the ideas in the original algorithm; it is not a word-for-word translation.
- **Q. Why is this a problem?**

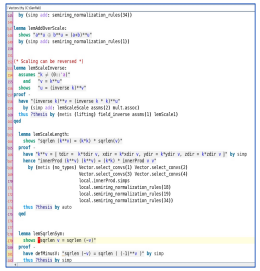


Do computer-generated works exist?

- Do computer-generated works exist which actually have 'no human author'?
- What if the programmer responsible for a computer generated work had some recognition of authorship under the CDPA?
  - This is problematic, because CDPA allows joint ownership of works.
  - A person who uses a computer program to produce output of commercial value could find the copyright owner of the program claiming part ownership of the output.
- The approach of the CDPA is effective but does not reflect the reality of the situation that all computer output is the result of human effort.
- One solution – make the programmer the joint **author** of a computer generated work, but ensure via the licence that the user is the **owner**.

Intermediate works

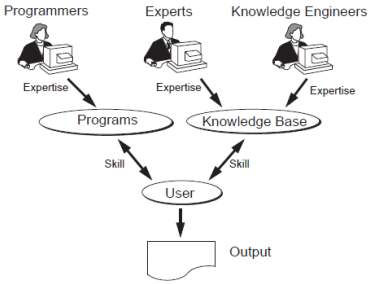
- Many works will arise as a result of the computer being used as a tool, e.g. a word processor.
- These works do not present a problem for the CDPA; the person who wrote the software package has no control over the content of its output.
- However, we can identify another category of **intermediate works** that lie between computer generated works and works made using the computer as a tool.
- The content of the output is the result of the skill and effort of the person using the software **and** the skill of the person who wrote it.



Example: *Using an automated theorem prover*. In this work (Stannett, 2019) parts of the proof are generated automatically, but built around a framework created using human effort.

Authorship/ownership of intermediate works

- Output is not entirely 'computer generated' since knowledge has come from a human author.
- The conclusion must be that there are **joint authors** of the system's output.
- If this interpretation is followed by the courts, it could lead to further legal problems, e.g. who is **liable** for the consequences of bad advice provided by the system?



Look and feel

- A **literal copy** of a program involves duplication of a substantial part of it without permission.
- What if two programs 'look' and 'feel' the same, even if the code is completely different?
  - Copyright protects expression, not ideas. However, it is generally accepted in English Law that expression can extend beyond a literal form.
- Where should the line between idea and expression be drawn?
  - This is not clear in the CDPA, but has been tested in several cases is the U.S. These cases are not binding on English courts, but U.S. courts often set influential precedents.

Whelan v Jaslow (1987)

- **1978**: Jaslow started writing a program for Jaslow Dental Labs, but gave up and hired Strohl Systems. They delivered the software in 1979, written in EDL and running on an IBM minicomputer.
- **Later**: Whelan left Strohl and acquired the rights to the software (Dentalab). Jaslow wrote a new version (Dentlab) in BASIC, running on IBM PCs. Jaslow's company sued Whelan for misappropriating Jaslow's trade secrets. Whelan countersued, alleging infringement of Dentalab copyright.
- **Decision**: Dentlab was substantially similar to Dentalab because its structure and overall organisation were similar. So copyright had been infringed.

Unintended consequences

- *The Whelan decision initiated a period of excessively tight protection, suppressing innovation, since almost everything other than the broad purpose of a software work would be protected. The only exception was where the functionality could only be achieved in a very small number of ways … Later the same year, [a court] cited Whelan when finding that the overall structure, sequencing, and arrangement of screens, or the "total concept and feel", could be protected by copyright.*

  https://en.wikipedia.org/wiki/Whelan_v._Jaslow
- **Q. Why the exception if the functionality could only be achieved in a very small number of ways?**

Computer Associates v Altai (1992)

- CA creating a program for scheduling computer jobs, which included a component called ADAPTER.
- Altai also created a scheduler. One of CA's employees moved to Altai taking ADAPTER source code with him and persuaded Altai to restructure their system. About 30% of the new system was taken from the CA source code.
- **Decision**: $364,444 damages awarded for infringement of copyright.
- Altai rewrote the offending sections from scratch using different programmers – judge held that new version **did not** infringe copyright.

The Test Applied in Computer Associates International v Altai (1992)

- The Court of Appeal criticised Whelan v Jaslow for getting too close to copyrighting ideas and suggested a three-step test for non-literal copyright infringement.

Abstraction

Filtration

Comparison

- This test reduces copyright protection for program structure, menus and interfaces. May be no golden nugget left after filtration.

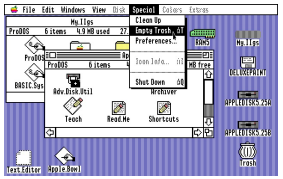
Identify the non-literal elements. This involves tracing the steps of the designer from the code back to the original program function.

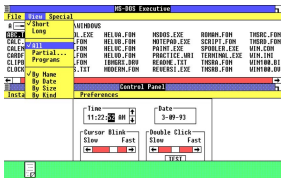
Some elements of the program (ideas, code dictated by efficiency, code in the public domain) are not protectable under copyright law. These elements are filtered out to identify the core (golden nugget) of protectable material.

Determine whether the defendant has copied a substantial part of the protected expression

Apple v Microsoft (1988-1992)

- Apple claimed that the 'look and feel' of Windows was based on Macintosh GUI
- Argued that individual elements of the design were not as important as the overall look and feel obtained by combining them.
- **Decision**: "*Apple cannot get patent-like protection for the idea of a graphical user interface, or the idea of a desktop metaphor*"





No look-and-feel protection?

- No protection for "look and feel"?
- "N*o principle can be stated as to when an imitator has gone beyond the copying the 'idea,' and has borrowed its 'expression'. Decisions must therefore inevitably be ad hoc. In the case of designs, which are addressed to the aesthetic sensibilities of an observer, the test is, if possible, even more intangible.*"

J. Learned Hand, ruling in
Peter Pan Fabrics, Inc. v. Martin Weiner Corp 1960

- In the case of Apple, an earlier licensing agreement that it had made with Microsoft was a key factor in the case, as well as its use of some materials from Xerox.

https://upload.wikimedia.org/wikipedia/commons/a/aa/XeroxWorkstation.jpg

Xerox Lisp machine (1985)

Non-literal copying (UK)

- **John Richardson Computers v Flanders (1993)**
- Richardson wrote a program, and later wrote another program for someone else to do the same thing. It was accepted that there was no deliberate copying.
- "*But the fact remains that he had …. an intimate knowledge of the [earlier] program at all levels of abstraction … and it is possible that he has, unconsciously or unintentionally or in some other way which he did not consider to be objectionable, made use of that knowledge … I find it impossible not to conclude that the line editor in the [later] program has substantially been copied from the line editor in the [earlier] program. If all that had been copied was the concept of a line editor that would not have mattered for present purposes, being a mere adoption of an idea. But similarities such as … the idiosyncratic restoration of text which is merely deleted and not replaced demonstrate that there has been copying of expression as well as idea*."

Ferris J, quoted at: https://swarb.co.uk/john-richardson-computers-v-flanders-chd-1993/

Summary

- The CDPA applies to computer programs and restricts acts such as copying and adaptation
- Applies to a whole or substantial part (test of quality, not quantity)
- Decompilation, making backups and error correction permitted under some circumstances by CCPR
- Aspects of the CDPA are controversial and raise potential legal issues, e.g. in relation to ownership of intermediate works
- Infringement on grounds of 'look and feel' is possible, but case law is inconsistent.