

## A New Era in Software Protection Law?

by Robert Goodman & Stephe Wilks

---

A recent United States Court of Appeal decision has been heralded as the leading precedent in the software copyright field and the beginning of a new era in software protection law (American Committee for Inter-Operable System - ACIS). Although it may be going too far to suggest that a new era has begun, it is reasonable to say that the clarity of the decision will greatly aid Australian judges and the IT Industry as a whole in formulating appropriate tests for the drawing of the line between protectable elements of computer software and non protectable.

The decision, in *Computer Associates -v- Altai* which was handed down by the US Court of Appeal on 22 June 1992, affirms the trial court's decision that Altai's computer program did not infringe the copyright in Computer Associates' program. The Court of Appeal agreed that the two programs were not substantially similar in respect of those parts of their expression which were considered protectable. The decision rejected a 1987 US decision in *Whelan -v- Jaslow* which provided protection for the 'structure, sequence or organisation' of a computer program.

The case was based on a claim by Computer Associates ('CA') that Altai had infringed its copyright in part of the CA program, CA-SCHEDULER. The particular component of the CA program, ADAPTER, was part of a computer scheduling package, and had the role of interfacing between the scheduling software and various operating systems.

The District Court found that OSCAR 3.4, Altai's original program, had infringed CA's copyright in ADAPTER

as approximately 30% of the source code had been directly copied by a former CA employee. However, OSCAR 3.5, which was substantially rewritten to remove the copied elements of ADAPTER, did not infringe CA's copyright. CA appealed from this decision.

The Second Circuit Court of Appeal upheld the District Court's decision and in doing so called into question a prominent US authority (*Whelan -v- Jaslow*). In the course of its review, the Court propounded a useful test for deciding whether copyright in a computer program has been infringed.

In *Whelan -v- Jaslow*, the Third Circuit Court of Appeal held that the **idea** of a utilitarian work, such as a computer program, is its **purpose and function**. The Court held further that where there were various means of achieving the same function, the method used to achieve this particular purpose and function is the expression of the idea.

The Court in *Altai* rejected the *Whelan* approach of examining a computer program as one idea. The Court took a more practical approach, regarding the overall program as a '*composite result of interacting subroutine*' each of which '*may be said to have its own idea*'. The Court laid down a three stage approach for assessing substantial similarity of computer programs which it proposed for future use by lower Courts faced with assessing whether copyright infringement of a computer program has occurred. The stages outlined are abstraction, filtration and comparison.

### **Abstraction**

At this stage, the Court would break down the allegedly infringed program into its constituent parts. This approach recognises that computer programs consist of a number of ideas and expressions. This process is described as '*reverse engineering on a theoretical plane*' and begins with a review of the code and ends with an articulation of the program's ultimate function.

The process first examines the source code, which is the lowest level of abstraction (where the structure may be quite complex). After determining the various procedures and sub-routines the process moves up through the various levels of abstraction and ends with an outline of the program's ultimate function (the highest level of abstraction). Along the way, a number of 'patterns' of increasing generality will apply to the components identified, corresponding to the decreasing protection afforded to the idea as opposed to the expression. In this way, the various **ideas** can be separated from the **expression** of those ideas in the computer program.

### **Filtration**

The second stage requires an examination of each part of the program in order to sift out those elements which are non-protectable, leaving behind a '*core of protectable material*'.

In carrying out this process the Court had regard to the merger doctrine. That is, where there is only one or a small number of ways of expressing an idea, the idea and its expression may be inseparable. In such a case,

copyright will be no bar to copying that expression since there can be no copyright in an idea. In computing terms this means that while there may often be a number of ways of carrying out certain functions within a program, external factors such as efficiency concerns and perhaps even the existence of a competitive market may limit implementation choices, especially in the case of utilitarian programs, to the extent that idea and expression merge.

The merger doctrine was approved briefly in the Australian High Court decision of *Autodesk -v- Dyason* in which Dawson J stated; '*When the expression of an idea is inseparable from its function, it forms part of the idea and is not entitled to the protection of copyright*'. The merger doctrine in relation to protection of computer programs has yet to have a full exposition in an Australian court. In particular, no Australian Court has reviewed the question of the effect of market pressures on the design or expression of a computer program (with an important example being the frequent implementation of 'Epson' compatibility in printer manufacture).

The Court also stated that other external factors which affect the way in which a program is written must be taken into account when sifting through the elements of the computer program. External factors noted by the Court include:

- ◆ mechanical specifications of the computer on which the program is intended to run;
- ◆ compatibility requirements with other programs;
- ◆ computer manufacturers' design standards;
- ◆ demands of the industry;

- ◆ widely accepted programming practices.

The final consideration mentioned by the Court is a determination of whether the program contains any public domain software, which is also non-protectable.

When the abstraction was originally carried out in this case, Pratt J, the judge at first instance, found that the OSCAR programs contained '*no protectable expression whatsoever*'. While the Court of Appeal states that perhaps he was being '*overly thorough*', it must be remembered ADAPTER was a low level interface between the more complex Application Software and the lower level Operating System Software. It is therefore not surprising that there would be very little in the program which was truly protectable given the external constraints imposed on the development of interface software of this nature.

It must also be noted that the Court of Appeal determined that the abstraction must be carried out on the CA program, not the allegedly infringing software. The Court took this approach as the abstraction process is used to determine the parts of the program in which CA can claim copyright. In doing this, the court distinguished between the scheduling component, which would have stronger claims to copyright protection, and ADAPTER which was the more utilitarian component of the software.

### Comparison

The final stage is the comparison of the remaining core of protectable expression with the expression used in the infringing program. At this stage, the inquiry focuses on whether the allegedly infringing software copied any aspect of protected expression of the original software.

### Conclusion

The three step approach proposed by the Court of Appeal is designed to address the policy behind copyright protection. That is, while affording protection to authors as an incentive to create new works, copyright must also be limited to the extent that its protection will avoid the effects of monopolistic stagnation. The Court of Appeal specifically stated that it sought it to insure two things:

- ◆ that programmers may receive appropriate copyright protection for innovative utilitarian works; and
- ◆ that non-protectable technical expression remains in the public domain for others to use freely as building blocks in their own work.

At present there are no satisfactory guidelines to assist Australian judges in deciding on the substantial similarity between computer programs. The method outlined above accords directly with the policy of copyright protection and may be applied under both the Australian Copyright Act and the Berne Convention. It is to be hoped that this approach, which will encourage the development of interoperable systems while still affording protection to system developers, will be adopted by Australian Courts and the IT Industry as a whole. ☞

*Robert Goodman & Stephen Wilks practise in the area of technology law, Gilbert & Tobin.*