



What Is Computer Software, And Does It Need Protection?



In this article, Alan Stuart, a partner with the Warkworth Lawlink firm of Webster Malcolm and Kilpatrick, discusses in depth the issues surrounding the protection of computer software against piracy.

In answer to the second part of this question one might argue as follows:

- computers are a valuable social and commercial tool, and it is beneficial for society that we continue to explore new and better ways of using them;
- this requires new software (among other things);
- software production is an activity that often demands considerable investment in time and money;
- if they are to have an incentive to develop new software investors must be able to obtain a return on their investment;
- without direct subsidies investors can obtain a return only if they are able to sell their product in sufficient quantities and at a good enough price;
- they have an opportunity to do this only if they can stop others from selling cheap reproductions of their work;
- although there are technological ways of preventing unauthorised copying, these methods can often be by-passed, and are not

by themselves sufficient. They need to be accompanied by legal sanctions, and privately enforceable rights are one way of giving software producers (like authors and inventors) the private means of protecting their markets;

- without ownership rights in their intellectual property, and a legal environment where they can enforce those rights, producers are not able to control the market sufficiently to obtain a return on their investment;
- unless producers are able to obtain a return they will abandon the market.

In New Zealand the principal means of protecting computer software against piracy is via the Copyright Act 1994, which in section 2 classifies computer software as a literary work. This means that new computer software has the same level of protection as a novel or play, ie protection against unauthorised copying for fifty years after the death of the author. In the modern world that level of protection seems excessive, and fails to recognise that conferring intellectual property protection (or any kind of monopoly protection) is inherently anticompetitive and is justifiable at law and in economic terms only because - and to the extent that - providing a limited monopoly gives an incentive to produce more work. But the other problem with the existing law is that it simply fails to effectively protect the valuable elements of modern software.

The analysis at the beginning of this article suggests that what software producers seek protection for are those elements of their software that require heavy investment in time, money and resources to produce. The trouble with our current regime - and this is a problem that has emerged only since our present Copyright Act became law - is that it is better at protecting the commonplace elements of software

that are now so universally available that they are no longer worth copying, than at safeguarding the aspects of the work that are truly novel and valuable.

With the increasing off the shelf availability of functional computer programmes it has become relatively easy and cheap for a software producer to compile a package that will perform the same tasks as the proprietary software produced by its competitors. What the software producer wants to develop and protect is the unique **look and feel** of its software, so that its customers looking for new software will choose something that is familiar (similar desktop appearance, key strokes, icons, sequences, methodology etc).

But these are precisely the elements that are least well protected by our existing laws, which are better adapted to protect the literal elements of both literary works and computer software, that is the expression of the software/story in letters and symbols. The extent to which copyright can protect the non-literal elements of a work (in the case of a novel or play the characters and plot, in the case of computer software the **look and feel**, and the design) is much less certain. In a number of decisions in Australia, the United States and England the courts have been reluctant to extend copyright to these aspects of software. The result is that new software that looks the same, and - from a user perspective - works the same, but at a technical level is different, did not infringe the original author's copyright. In some ways this outcome will be a surprise. But it stems from the origins of copyright as a means of protecting the expression of ideas, but not the ideas themselves.

When the software language was common to most systems, where the approach to writing it was uniform, and where there were relatively few with the special skills required to write software, it may have been feasible to rely on a

system for enforcement that depended on originality of expression. The structure of the software industry in its early days meant that competing software systems were easily compared to see if they were likely to have been copied. The smaller numbers and larger size of the participants in the industry meant that they had the resources to enforce their property rights, and had something to lose if they were found to be infringing. The medium that software was written in was called a **language**, and it was only natural that copyright principles that protect the use of language in expression should also be used to control this new industry. But if software is now written/assembled, sold and used in a different way from what it was in the 1980s when the current system of protection was created, should we not consider whether there are methods of protection that will better meet the current needs?

Furthermore, the increasing difficulty that authors have in establishing copyright in those elements of computer software that they want to safeguard, suggests that the protection that literary copyright confers is illusory. Software producers do not need the level of protection currently offered by copyright for those elements of software that are covered by that protection. But they are unable to obtain protection for those aspects of their work that are most valuable to them.

There seem to be four alternative means of protection: patents, the law relating to compilations, the law of design, and special legislation dealing only with computer software. But the first three suffer from the same problems in adapting them to computer software that plague the copyright solution we have at present, while special legislation requires a degree of academic and political consensus that does not seem achievable.

Is there a need for any solution? After all, we have **muddled through** up till now with a combination of copyright law, patents, a bit of specialist legislation and contractual and technical solutions. This is the same combination of laws and other methods of control that apply to most other fields of commercial activity, for which there has been

no call for specific legislation. We don't have special legislation for books, movies, or the car industry.

When we come to assess what protection is necessary, we should do so by referring first to the original concept of copyright; as a privilege that is justified only if and to the extent that it is in the public interest. It is widely accepted that some protection is essential to ensure further creativity and inventiveness, but there is little proof of that assertion, and no way of knowing at what point the losses from conferring protection match or outweigh the gains. In the arena of computer technology, the history of open source software and the Linux operating system suggest that the assumption about the public benefits flowing from protection may be illusory. In the case of those areas of software development progress seems all the more vigorous **because** (rather than in spite of the fact) there is no attempt to claim private property rights over the concept. If the courts are reluctant to extend the concept of non-literal infringement of copyright is that necessarily a bad thing?

It is easy and perhaps natural to regard the problems of the present as being more significant than any that have arisen in the past. It can also be dangerously complacent to say **wait and see**. But in the absence of an obvious problem, let alone an obvious solution, that may be the best advice for now.

Email a.stuart@wmklaw.co.nz
Website www.wmklaw.co.nz