indication of infringement. To apply this level of abstraction to computer programs is highly inappropriate (it may for instance result in one word processing program infringing a second on the basis of function alone), and when combined with the Court's decision that individual words may constitute a computer program, a conclusion of infringement was inevitable.

Implications

The decision poses great difficulties for software developers in Australia. Local developers, unlike their counterparts in the US and Europe, may not it seems reverse engineer, let alone decompile, another product in order to understand the unprotected ideas and produce a compatible, interoperable product (particularly if

the product created performs the same function as the original).

This amounts to the death knell for open systems and means that in Australia the pace and scope of innovation will be controlled by foreign companies whose product interfaces become industry standards. Open systems development will continue, but not in Australia.

Recommendations

Even though Powerflex had indicated it will appeal the decision, there is a need for legislative amendment to the Copyright Act in the following areas.

1. The adoption of a provision equivalent to s102(b) of the US Act to exclude methods of operation from copyright protection. This amendment would provide

guidance to the judiciary in cases such as this, allowing them to draw upon the extensive experience of the US decisions. At the same time such an amendment would bring Australia into compliance with its outstanding obligations under the TRIPS Agreement.

2. An amendment to the definition of computer program in line with the recommendations of the CLRC. Such an amendment would avoid the current emphasis on function evident in the Powerflex decision and the earlier High Court case of Autodesk v Dyason.

Ashley Porter is Legal Counsel at Sun MicroSystems in Sydney.

1   (1006) 33 IPR 194

Data Access Corporation v Powerflex Services Pty Limited

Ron Eckstrom

It was only in 1983 that WIPO concluded that it was premature to recommend an international model by which computer programs should be protected.[1] Since then world opinion has effectively crystallised and copyright is the preferred means of protecting computer programs. Article 10 of TRIPS specifically extends the Berne Convention to recognise computer programs, whether in source or object code, as literary works. More than anything else, the need to adequately protect the significant investment in creating software has led to this resolution. Software is an essential element of the information economy and few businesses could operate efficiently today without information technology, all of which requires computer programs to effectively function. The success of the software

industry and the flow on effects are due in no small measure to copyright protection for software. From an early stage, Australia has recognised this commercial reality. It is only necessary to point to the 1984 amendments resulting from the first instance decision in the *Apple v Computer Edge* litigation to demonstrate this.[2]

Some of the most recent figures and forecasts relating to Australia's information industries make it clear that copyright protection of software is of great economic benefit to Australia. The Productivity Commission's report *Mapping the Information Industries* shows that during the period 1990-1994 software royalty receipts grew by 44% on a compound annual growth rate basis while software royalty payments grew by only 6% over the same period.

In general, the principles of copyright law may be applied to computer programs with no difficulty. However, from time to time the unique characteristics of computer programs raise complex questions for copyright law. In particular, the idea/expression dichotomy, although it is a central tenet of copyright law, is often difficult to apply. The utilitarian nature of computer programs has meant that much of the refinement of this fundamental principle has occurred in cases involving allegations of infringement of copyright in computer programs. These complex legal issues have been considered in the context of highly technical evidence - something which the participants in the legal system, both judges and lawyers, often find difficult to absorb.

The two well known Australian cases examining subsistence and infringement of copyright in computer programs are *Apple v Computer Edge* and *Autodesk v Dyason*. In both cases it is clear that the courts had difficulties in interpreting the facts and applying existing copyright principles to them. In particular, the decisions at first instance bore little resemblance to the ultimate judgments of the High Court. Now we have another prominent software copyright case: *Data Access v Powerflex*. Just like the Apple and Autodesk cases, the initial decision has created a furore. It is now on appeal to the Full Federal Court. Given the history of the *Apple* and *Autodesk* cases, whatever the result of the appeal we may expect that the reasoning of the Court may be quite different from that of the trial judge.

There can be little doubt that many were surprised to learn from the first instance decision in *Powerflex* that a reserved word in a programming language could be a computer program in its own right. However, the long term implications of the decision are unlikely to rest on this aspect. Rather, the case is most important in its treatment of the idea/expression dichotomy in so far as it affects the ability of developers to create interoperable systems.

Interoperability is accepted within the software industry, worldwide, as a desirable goal. Interoperable products benefit the copyright owners on both sides of an interface. Particularly in the field of operating systems software, open systems are critical. Many of the major software houses have recognised that the marketability of their products depends on independent developers being able to write software which successfully interfaces with those products. Microsoft's Developer Network Program is an example of the action taken to encourage software developers to create interoperable products. IBM and Novell have similar outreach programs. The aim of each is to make available to developers information and tools to foster the development of software that is interoperable with existing

software platforms. In this case, Powerflex desired to *independently* create a programming language which could manipulate files created with the DataFlex software. From a policy perspective such activities ought to be encouraged.

Amongst other matters, the case focussed on an error text table (used by both programs to generate syntax and other error messages), a Huffman compression table (used by both programs to compress the data stored in a database created through the use of both programming languages) and the set of instructions determining the "file structure" of the databases created through the use of both programming languages. In particular, the Huffman table and the file structure were critical to Powerflex's goals of achieving interoperability between databases created through the use of DataFlex and PFXplus.

In each instance there are two arguments to be considered: whether the subject matter is copyrightable; and , if so, whether a reproduction or adaptation in fact occurred. Powerflex's argument was that each of these items was not copyrightable because their function was inseparable from their expression. In respect of the error text table this argument was successful. However the Court held that the Huffman compression table was copyrightable and that a reproduction had occurred. The Court did not accept Powerflex's submission that the function of the table was to achieve compatibility with the DataFlex programming language and that there was only one way of expressing that function. Rather, the Court chose to abstract the function of the Huffman table to a higher level, namely compression of data. On that basis, it was open to Powerflex to adopt a range of different parameters in the Huffman table.[3]

It is difficult to comment on the aspect of the decision concerning infringement by Powerflex of the set of instructions determining file structure of the DataFlex programs. The decision does not describe these

instructions in any detail. However, based on the scanty factual material available, the reasoning of the Court appears to come close to the opinion of Northrop J in the first instance decision of *Autodesk v Dyason*, which was forcefully reversed by the High Court, that functional equivalence constitutes reproduction.

Given the appeal, it is too early to conclude whether the *Powerflex* case should prompt legislative change. However, one thing is clear, and that is that the case should not been seen as a justification for liberalising the restrictions on decompilation of software under the Copyright Act. It must be recognised that the case is not about decompilation. Data Access believed that Powerflex may have created the PFXplus software by means of decompilation. However, at the trial it was found that no decompilation had in fact occurred. Dr Bennett's evidence, which was accepted, was that he developed the PFXplus software through close observation of the behaviour of the Dataflex product.

As stated above, interoperability is an important goal and all participants in the software industry will be hoping that the Full Federal Court's decision will provide a framework that will continue to encourage the independent development of interoperable products.

Ron Eckstrom
Asia Pacific Counsel Microsoft

1   WIPO, "Legal Protection of Computer Software" (1983) 17 *J World Trade Law* 537 at 544-5.

2   See McKeough and Stewart, *Intellectual Property in Australia*, Butterworths 1991, pp 172-173.

3   There is another argument not addressed in the judgment on which it may have been possible for Powerflex to avoid a finding of infringement in respect of the Huffman table. Once it is accepted that the table is copyrightable, it does not follow that reproduction did in fact occur. Counsel for Data Access argued that the Huffman table was like a logarithm table - it required substantial skill to create. However, it is possible that two people could independently create two identical logarithm tables using the same methods. In that case, the creation of the second table would not be an infringement of copyright in the first. That may in fact have been the case in respect of the Huffman table here. However, the judgment does not deal with such an argument.