The law of Singapore has yet to deal with any of the new and difficult issues peculiar to the protection of intellectual property rights in computer programs, but it is probably only a matter of time before it does. When that time comes, we shall have the luxury of being able to learn from the experience of other jurisdictions. It is proposed in this article to briefly identify the major legal issues peculiar to the intellectual property protection of computer programs and consider developments over the last couple of years with an eye to the lessons that they hold for us. Constraints of space means that consideration will be limited to landmark cases or statutory interventions and trends in the law that have become discernible over that period in the United States, the United Kingdom and Australia.

I. INTRODUCTION

a. Computer Programs

Before considering developments in the law, it is necessary to introduce the subject matter of this article briefly. A starting point would be the definition in the Section 7 of the Copyright Act:

“computer program” means an expression ... of a set of instructions ... intended ... to cause a [computer] to perform [functions].

This definition would almost certainly drive a technically trained person to despair for its lack of technical precision, but it does indicate two very important things about computer programs:

(a) every computer program is essentially a set of instructions to the computer on which it is to run; and

(b) when a properly-written program is run on a computer, the computer performs functions. Without computer programs to direct the computer’s internal operations, the computer hardware is nothing but inert machinery.

These two features are the source of a great deal of conceptual and legal difficulty. Computer programs are ‘information’ (that is, sets of instructions)
and yet at the same time, they are akin to machine parts. Computer programs are intangible but are capable of leading to tangible physical results.

Although we tend to credit computers with almost superhuman ‘intelligence’, the truth is that the brains of a computer, its central processing unit (‘CPU’), is capable of performing only a few very basic logical operations. The magic behind the computer is that it can perform these operations at great speed under the control of a computer program. It is the manipulation of electronic inputs at great speed by the CPU and other hardware in accordance with the program that produces electronic outputs which may in turn result in the tangible physical results that we now take for granted. The detailed technical wizardry by which this is achieved lies beyond the scope of this article, and the writer’s ability to explain briefly, so we will have to take this effect as a given. Those who wish to explore the technical aspects of computers and computer programs further may wish refer to sources intended specifically to explain the basics of computer technology to the layman.

b. Copyright

The law of Singapore and the other countries under consideration provides that the basic protection for persons who develop or exploit computer programs is through copyright. The choice of copyright is a pragmatic one; the other major forms of intellectual property protection — patents, the law of confidence and trademarks — are in many ways even less suitable for dealing with computer programs. This choice has been implemented in all the jurisdictions under review by simply treating computer programs as literary works. This means that effectively, intellectual property protection of computer programs focuses on the ‘instructional’, textual aspect of programs rather than its ‘machine part’ aspect. On the surface, that is understandable. Every computer program is capable of taking a form very much like a traditional literary work; the instructions constituting the program may be printed out in ‘hardcopy’. Stored in a computer, those

3 The instructions that make up a computer program can be embodied in the circuitry of a computer chip such as a ROM (‘Read Only Memory’) chip so computer programs can take the form of a physical device. The flexibility of the form which a program can take does not detract from the fact that what is important about the program are the instructions which make it up. A program in the form of text written on a floppy disk may have exactly the same functionality as a program embodied in a physical device.

4 A series of books by the Ziff-Davis Press presents basic concepts of computing in a very accessible form; see White, How Computers Work (1993) and How Software Works (1993), and Derfler & Freed, How Networks Work (1993).

5 Singapore removed any doubt that the primary means of protecting computer programs was by way of copyright in literary works by the simple expedient of amending the definition of the term ‘literary work’ in Section 7 to include computer programs and compilations of computer programs.
instructions are little different from any word processor file which might contain a true literary work. The normal use of such a program simply involves its translation into electronic signals for electronic consumption by a computer, and all of this seems well within the ability of copyright law to cope.

At this level, copyright law is more than adequate for dealing with ‘piracy’ of computer programs — mass, unauthorised copying of entire programs for sale at prices that undercut legitimate copies. In this respect, computer programs are no different from other copyright works like compact discs and video cassette tapes since the copying involves taking the entire content of the work, or of a computer-usable translation of the work. The problems in this area are the same practical problems of enforcement and proof that afflict other copyright works and thus fall outside the scope of this article. The problems unique to the application of copyright to computer programs lie elsewhere.

At the risk of oversimplifying, the similarity between computer programs and normal literary works like novels goes beyond the fact that computer programs can take the form of words on a page. Besides the literal text of the instructions, which corresponds to the words of a novel, a computer program has a textual ‘structure’ which can be compared to the ‘plot’ of a novel. The sequence of instructions making up the program must be organised according to programming rules to achieve the ultimate purpose of the program, just as the words making up a novel have to comply with rules of grammar and logic in order to effectively communicate the meaning which the writer wishes to convey.

This apparent similarity is deceptive; copyright protection of the structure of computer programs presents a number of special difficulties which do not arise with traditional literary works. First, the structure of a computer program is likely to be dictated by external factors to a much greater extent than the plot of a novel. Considerations like the function to be performed by the program and the dictates of the hardware on which it is to be run have a very significant impact on the form which a program’s structure ultimately takes. A computer program must be as ‘efficient’ as possible to be commercially viable; it must not occupy too much space in a storage medium like a computer hard disk, it must be able to lead to the computer’s performance of the appropriate functions at a reasonable speed given the limitations of the CPU and the random access memory (‘RAM’) which the user’s computer is expected to contain. Much of the commercial value of a computer program lies in its structure, since this determines how

---

6 A discussion of the process of programming and the constraints on programmers is to be found in an article by PS Menell, “An Analysis of the Scope of Copyright Protection for Application Programs” (1989) 41 Stan L Rev 1045; This article proved to have some influence on the major US authority in this area; see the main text accompanying note 27, infra. Cf A Clapes, Software, Copyright & Competition (1989).
well the program deals with these constraints to operate as the user would demand. In this regard, the text of the program is far less significant than its structure.

Also, much of the effort and cost of designing a computer program goes into the design of the program structure, and the writing of the literal text is again relatively insignificant in comparison. Non-literal copying of programs — that is, copying the structure of the program only — is thus a commercially viable activity because a programmer writing a program to perform a particular function could save himself much time and effort if he simply copied the structure of a successful program which performed a similar function. This would hold true even if he had to supply all the literal code himself. Copyright law is fundamentally flawed in its application to computer programs because it focuses on the literal text of the copyright work in question, which in the case of a computer program is often relatively less valuable than the structure.

While copyright does provide some protection for non-literal aspects of traditional literary works such as the plots of novels, its application to the non-literal aspects of computer programs is conceptually more difficult than for traditional literary works. It is a relatively simple matter for a judge faced with a claim that the plot of a novel has been copied to compare the two works and intuitively decide whether ‘copying’ of plot is so significant that it amounts to infringement. Such analysis is not possible with computer programs since few judges are in a position to perform that kind of intuitive analysis. Judges hearing claims for copying of the structure of computer programs are therefore faced with the problem of putting into clear words the appropriate legal analysis so that experts can reliably assist in the process.

The constraints affecting computer programs add another difficulty in relation to claims for infringement by copying of the structure of a program. Proof of copying of copyright works usually involves showing such close similarity between the alleged copy and the allegedly copied work as to make independent creation unlikely. But programs performing similar functions on similar systems will be subject to the same set of external constraints, so there is a greater likelihood of similarities between two such works, even if they were independently created. It would therefore be more difficult in practice to establish copying on the basis of similarities between two works which are subject to the same constraints since those similarities might also be a result of independent creation subject to the same external constraints. In a literary work, the problem would be comparable to trying to determine whether copying had taken place where two writers wrote descriptions of the same historical subject, with both under a very strict word limit as well as the same restrictions on the range of vocabulary that could be used in the description. Similarities between the two resultant works would not necessarily be probative of copying. If the constraints were severe enough that there was only a very limited
number of ways in which the subject could be described, a further question might arise: whether copyright could subsist in such a work at all. This is a distinct problem from the problem of proof described earlier, and will be referred to hereafter as the problem of ‘merger’ of expression of a program with its function, adopting the terminology used in American cases.  

‘Structure’ of a computer program has another, even more complex dimension. It also has a ‘behavioural’ aspect — the way the program behaves when it is run in a computer. The various instructions making up the program text are run in the computer’s CPU in sequences determined by the program itself and sometimes by the user. Data (which may have been newly input by the user or simply data which is stored within the program itself or in discrete files that are not part of the program) is manipulated and moved from one memory storage area to another within the computer and from memory storage area to CPU and back while instructions are sent to and received from peripheral devices. Again, all this occurs in a manner determined by the program and sometimes by the user. The manner and sequence in which all this happens is a kind of ‘structure’ (‘behavioural structure’ hereafter) which is alien to traditional copyright law, and it is perhaps for this reason that it is often overlooked in computer program copyright cases which deal with copying of structure. An exception to this trend is the aspect of behavioural structure called the ‘user interface’.

The term ‘user interface’ is not a term of art, but typically encompasses the means by which the computer program presents information to its user and the user gives instructions to the computer program to direct its operation. The user interface thus consists of visual and aural information presented by the program to the user, as well as the means by which instructions are issued by the user to the program: the designation of keystrokes for particular commands, or perhaps use of a mouse or other pointing device in conjunction with specially designed screen menus. Given that a good user-interface can have a significant impact on the popularity (and thus the commercial success) of a computer program, it is natural that copyright-owners have claimed that this too comes within the scope of copyright protection, however conceptually unsuitable that regime may be.

These copyright problems have received considerable attention in the District and Circuit Courts of the United States of America over the last few years so discussion there has reached a high level of technical and legal sophistication. Cases on this problem have only just begun to emerge in the United Kingdom, and these have made reference to the pioneering experience of their US counterparts, so it will be necessary to examine copyright developments in America first. Our consideration of copyright law in the UK will focus mainly on the attitude of the UK courts to

7 The doctrine of merger in US copyright law is discussed at note 35, infra.
8 See the main text accompanying notes 22 to 61, infra.
American authorities, and also the new UK Regulations on protection of computer software which modify the rules of copyright in relation to computer programs.

c. Patents

Although copyright is considered the most appropriate right for protection of computer programs, patents are sometimes sought for inventions or processes which consist in part of computer programs.

Singapore is still in the process of enacting its own Patents Act, so it would be inappropriate to consider the basic rules of patent law in detail. The Patents Bill received its Second Reading in Parliament on the 21st of March 1994. If all goes well, we will have the legal framework for a local patent system in place before the year’s end. Implementation of the system may take a little longer.

The basic principles of patent protection under the Bill do not vary significantly from those applying in the other jurisdictions under review; an invention will only be patentable where it is novel, it involves an inventive step and is capable of industrial application. The first two requirements represent significant obstacles to the patentability of computer program related inventions, but this problem is not unique to computer technology, so it will not detain us further.

Our attention will instead focus on the stumbling-block to patent protection which is specific to computer programs — Clause 13(2) of the Bill, which reads:

> It is hereby declared that the following (among other things) are not inventions for the purposes of this Act, that is to say, anything which consists of ...

> (c) a scheme, rule or method for performing a mental act, playing a game or doing business, or a program for a computer ...

but the foregoing provisions shall prevent anything from being treated as an invention for the purposes of this Act only to the extent that a patent or an application for a patent relates to that thing as such (emphasis added).

---

9 See the main text accompanying notes 82 to 105, infra.
10 The UK Copyright (Computer Programs) Regulations 1992, S.I.1992 No.3233. This will be discussed in the main text accompanying notes 70 to 81, infra.
11 Readers wishing to acquaint themselves with the basic rules of patent law as it presently stands in Singapore may wish to refer to SK Ng “Intellectual Property Law in Singapore” (1992) 4 SAeLJ 32.
12 Bill No 4 of 1994.
13 These requirements are set out in clause 13(1) of the Singapore Patents Bill.
In other words, if the invention consists of nothing more than a computer program running on standard equipment, it is ineligible for patent protection. But if a computer program is only one integer in the invention, then that invention is not necessarily ineligible for patent protection.

This formula very closely resembles that used in Section 1 of the UK Patents Act 1977\(^\text{14}\) and there are a number of cases in the UK which have had to wrestle with the implications of that awkward phrase ‘that thing as such’. Those decisions would thus be relevant to the interpretation of our own provision (if it should be enacted unchanged) and the most recent of those cases will be considered in the section on developments in the UK\(^\text{15}\). Parties considering UK authorities must, however, bear in mind that since the accession of the UK to the European Patent Convention of 1973, UK courts have considered the terms of that Convention and decisions of the European Patent Office as aids to interpreting the UK Act\(^\text{16}\). There is much to be said in favour of Singapore’s patent system developing (when it does) in step with the patent laws of other jurisdictions such as the parties to the European Patent Convention. This would be achieved if local courts adopted UK authorities as far as possible. But UK authorities are not strictly binding and care must be taken to ensure that the local patent law does not become confused with concepts or policies that make sense only in the context of the European Patent Convention through uncritical adoption of the reasoning in those cases.

It is also worth noting that even though US and Australian patent legislation differs from the scheme adopted in our Patents Bill in not expressly excluding computer programs ‘as such’ from being patented, case law in both those jurisdictions has addressed similar concerns as have led to the Clause 13(2) of the Bill. Recent developments in those countries will thus also be considered for their relevance to the development of Singapore’s own patent law.

II. RECENT DEVELOPMENTS: THE UNITED STATES OF AMERICA

Although the rules of copyright and patent law in the jurisdictions that make up the United States of America are significantly different from ours, the basic concepts on which these two sets of rules are built are similar enough that cases decided there are worth examining. This is particularly so because the US Courts have been dealing with more computer-related litigation for longer than the other countries under consideration — no doubt a result of America’s position at the centre of the information technology explosion. As a result, the law on copyright

\(^{14}\) 1977, c 37
\(^{15}\) See main text accompanying notes 111 to 127, infra.
\(^{16}\) Gale’s Application [1991] RPC 305 at 321–323.
and patent protection of computer programs may be converging around principles that may eventually apply consistently throughout the US.

a. Patent Law

US cases have shown a tendency towards a liberal determination of patentability of computer program related inventions. This attitude was expressed by the courts most clearly in the US Supreme Court decision in Diamond v Chakrabarty\(^\text{17}\) where it was suggested that “anything under the sun that is made by man” should be patentable unless legislation clearly excludes it. As such, it is relatively easier to obtain patents for ‘inventions’ that include, or in some cases consist largely of computer programs than in the other jurisdictions under review.

A recent case in the Federal Circuit, Arrhythmia Research Technology v Corazonix Corporation\(^\text{18}\) is worth reading for its review of the current position in the US and in particular for the judgment of Judge Rader who spoke in favour of an even more liberal approach to patentability of ‘algorithms’ than was previously thought to be appropriate. Judge Rader argued that the preoccupation of courts considering computer related inventions with ‘algorithms’ as unpatentable subject matter was misplaced. He questioned the assumption that any invention that consisted of only an ‘algorithm’ was unpatentable and argued that only mathematical procedures for solving purely mathematical problems ought to be denied patent protection\(^\text{19}\).

The approach of the US courts is reflected in the practice of the US Patent and Trademarks Office, which grants patents quite liberally to claims reciting computer programs\(^\text{20}\). A salutary warning of the dangers of this liberal approach was sounded by the controversy over a patent awarded to Compton’s New Media for a basic multimedia system. That award, it was argued, would have effectively given Compton’s a monopoly over multimedia products. The controversy that followed its announcement led to its reconsideration by the US Patent and Trademarks Office. The patent was ultimately revoked earlier this year\(^\text{21}\), but the scare which it spawned could lead to a re-evaluation of the appropriateness of the patent regime for

\(^{17}\) 447 US 303 (1980). This case did not involve a computer program related invention, but the attitude expressed in that judgment has clearly influenced later decisions on such subject matter.

\(^{18}\) 958 F2d 1053 (Fed Cir 1992)

\(^{19}\) Ibid, at 1064 and 1065.

\(^{20}\) For instance, the trend towards greater use of patent protection for software related inventions in the financial services sector is described and discussed in MJ Lennon, “United States Patent Rights in Financial Services Software” (1994) 10 CL&P 17.

\(^{21}\) The controversy and its resolution are described in some detail in Lennon, “The controversy over Compton’s multimedia patent” (1994) 10 CL&P 66.
computer software products within the US, and should certainly be borne in mind if consideration should ever be given to relaxing the prohibition against patenting computer programs ‘as such’.

b. Copyright Law

(i) Computer Associates v Altai

The approach of courts towards analysing the copying of the structure and other non-literal aspects of computer programs in the various jurisdictions of the United States appears to be converging around two essentially reductionistic approaches. Most attention has been paid to the approach adopted in Computer Associates International Inc. v Altai Inc. Before we can consider that decision in depth, it is necessary to briefly review the legal developments leading to that case.

The basic problems under consideration in the cases leading to Altai are the problems of applying copyright to the protection of structure of a computer program discussed earlier, namely:

(a) to what extent should the structure of computer programs be protected by copyright, given the close connection between the structure and the functionality of the program?

(b) What is the appropriate analysis to determine if infringement has taken place?

These two questions were not actually identified separately in any of the cases under consideration, but were dealt with together as part of the larger question of determining whether there was ‘substantial similarity’ between works which were the subject of a claim for infringement by non-literal copying. The infringement claimed in these cases was essentially copying of the structure of the program rather than of the literal text of the program code, and the courts dealt with these claims by adapting the analysis typically used for determining infringement by non-literal copying of traditional literary works. This involved asking whether the allegedly infringing and the allegedly infringed works were ‘substantially similar. A finding of ’substantial similarity’ between the works is generally determinative of infringement.

US courts determine whether non-literal similarities between traditional works (such as similarities between the plots of novels) are ‘substantial’ by attempting to distinguish the ‘idea’ of the work from its ‘expression’; a very general description of the work is its ‘idea’, while the literal text and perhaps a detailed outline of the plot is ‘expression’. Similarity of ‘ideas’ alone is

---

23 See the main text accompanying notes 6 to 8, supra.
never ‘substantial’ and will never amount to infringement while similarity of expression may be ‘substantial’ and thus may lead to a finding of infringement. The distinction between ‘idea’ and ‘expression’ to denote protectability by copyright is not peculiar to US copyright law, but it is cited much more readily in judgments of the US courts as a basis for decision because it is specifically provided for by statute.\(^{24}\)

This approach of distinguishing idea from expression was applied to the analysis of substantial similarity between computer programs by the US Court of Appeals for the Third Circuit in *Whelan Associates Inc. v Jaslow Dental Laboratory Inc.*\(^{25}\) The test for distinguishing unprotectable idea from protectable expression in computer programs was a simple and elegant one:

> [T]he purpose or function of a utilitarian work [like a computer program] would be the work’s idea, and everything that is not necessary to that purpose or function would be part of the expression of the idea ... Where there are various means of achieving the desired purpose, then the particular means chosen is not necessary to the purpose; hence, there is expression, not idea.\(^{26}\)

The test was immediately and rightly criticised as being overly simplistic and based on a misunderstanding of how computer programs worked\(^{27}\). It was not adopted by some cases prior to *Altai*\(^{28}\) but it was only in *Altai* that a serious attempt was made to formulate an alternative test which took

---

24 Section 102(b) of the US Copyright Act (Title 17 USC) provides: In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle or discovery, regardless of the form in which it is described, explained, illustrated or embodied in the work. Although most of the US cases have referred to the distinction between ‘idea’ and expression, the distinctions between ‘procedure’, ‘system’, ‘process’ or ‘method of operation’ and expression are probably equally important considerations for the copyrightability of computer programs since they are more closely related to the functionality of programs. Much of the time, reference to the ‘idea’ of a computer program is actually reference to these other types of unprotectable subject matter.


26 Ibid, at 19.


particular account of the special characteristics of computer programs. The decision in *Altai* will already be familiar to many but its importance requires a brief explanation of the test propounded in that case.

The recommended analysis of substantial similarity occurs in three steps, called ‘Abstraction’, ‘Filtration’ and ‘Comparison’.

The ‘**Abstraction**’ stage calls upon the trier of fact to analyse the program and identify the layers of abstraction within it. At the risk of over-simplification, one may see this exercise as requiring the identification of the layers of structure in the program; the text of the program is the ‘lowest’ level of abstraction, a description of the code is the next higher level, a description of the major components of that description is the next, and so on up to the basic statement of what the program does, which is the ‘highest’ level of abstraction. Judge Walker, who delivered the judgment of the Court of Appeals in *Altai*, described this exercise as the reversal of the normal process of writing a computer program, which typically begins with the computer programmer considering what the program is to do, then breaking down the identified task into a logical sequence of sub-tasks which are then further broken down into smaller and smaller sub-tasks until all these resultant sub-tasks are of the type which the computer is capable of performing. The final step is the writing of code to implement the detailed structure that has been worked out. It is perhaps an implicit part of this ‘abstraction’ stage that the behavioural structure of the program would also be identified, but this could be easily overlooked in the judge’s description of the test.

This stage essentially involves the recognition that ‘structure’ of a computer program is too complex to consist of only one ‘idea’, as the *Whelan* test assumed. Structure can be identified at a number of levels of abstraction, from a general statement of the major functional aspects of the program to a very detailed description of all the individual instructions or groups of instructions making up the program. If one examines the test as a whole, the value of having an ‘abstraction’ stage is essentially to remind the trier of fact that the structure of a computer program is so complex, existing as it does at a number of levels of abstraction, that the idea/expression dichotomy is meaningless unless one considers each level of abstraction in isolation. The ‘idea’ of a module becomes ‘expression’ when it falls to be considered at the next higher level of abstraction (which would tend to be a description of those lower level modules by their ‘ideas’). As a prescription, this stage suggests that the trier of fact must acquaint himself with not

---

29 It is discussed locally in Ng-Loy Wee Loon, “Reverse Engineering of Computer Programs — Rethinking its Prohibition” (1994) 6 SAC LJ 131

30 Supra, note 22 at 400 and 401.
only the literal text of the program code but also consider each and every level of structure from the most detailed to the most general\(^\text{31}\).

Once the layers of structure in the allegedly infringed program have been identified, ‘filtration’ occurs; the trier of fact must examine the individual elements of the program at each level of abstraction to see whether, taken in isolation, they are eligible for copyright protection or not. Elements which are unprotectable are filtered out leaving behind a ‘core of protected material’\(^\text{32}\). The unprotectable elements to be filtered out are those which are:

‘idea’, or [expression] dictated by considerations of efficiency, so as to be necessarily incidental to that idea; required by factors external to the program itself; or taken from the public domain...\(^\text{33}\)

Space prevents detailed discussion of what these classes of unprotectable expression entail and why they should be filtered out\(^\text{34}\). It is, however, worth pointing out that the court’s explanation of the category ‘expression dictated by considerations of efficiency, so as to be necessarily incidental to that idea’ is significant in that it deals with the problem of ‘merger’ of function with expression which is especially problematic for copyright in computer programs\(^\text{35}\). The existence of this factor in the Altai test recognises

\(^{31}\) Judge Walker, who delivered the opinion of the Court in Altai, suggests that this step is ‘conceptual’ while the next step, filtration, is ‘concrete’. It appears that the judge recognised that the first stage actually contains little by way of practical prescription, and merely directs the trier of fact to bear in mind the complexity of the program’s structure when he performs the other steps of this analysis; supra, note 22 at 401.


\(^{33}\) Ibid.

\(^{34}\) They are discussed at length in the judgment in Altai; ibid at 401 to 405.

\(^{35}\) Ibid, at 402. The problem of ‘merger’ is explained in the main text accompanying note 7, supra. The doctrine of ‘merger’ in US copyright law essentially provides that when the expression of an idea is so significantly constrained by the nature of the idea which it is describing, then copyright cannot subsist in that expression. In such a case, the expression is said to have ‘merged’ with the idea. This doctrine is based on the logic that copyright cannot subsist in ideas so it may similarly not subsist in expression that had ‘merged’ with its idea. To confer copyright protection on expression that had merged with idea would amount to conferring a monopoly on the idea; see Herbert Rosenthal Jewelry Corp v Kalpakian 446 F2d 738 at 742 (9th Cir 971). This logic is not compelling, however. Where expression is dictated by the underlying idea or functionality, conferring copyright protection on that expression would not amount to conferring a monopoly on the idea. Copyright law does not penalise independent derivation of expression, even if that independently derived expression were identical to the protected work. In a case where merger is complete (ie where there is only one way of expressing an idea), it would be impossible for the copyright owner to prove that his work had been copied rather than independently created unless he had clear evidence to disprove independent creation. Effectively, copyright could subsist in such a work, but would be meaningless since infringement would be practically impossible to establish. But this is not the same as asserting no copyright subsists in that work, as did the court in the Herbert Rosenthal case.
that efficiency is a serious constraint on the computer programmer’s expression. The *Altai* court also recognised that the doctrine of merger applied to individual elements within the program and was not limited to the overall ‘structure’.

The final step in this analysis is ‘**comparison**’ of the core of protected material left after filtration with the allegedly infringing program. The aim of this exercise is to determine whether the alleged infringer copied any protected expression, and to assess the ‘substantiality’ of that copied expression. If a substantial part of the protected expression has been copied, then infringement is established.

**Another Approach**

The *Altai* test is not the only analysis in the US for determining whether similarity between non-literal aspects of computer programs is ‘substantial’. Anyone considering US authorities on this point is likely to encounter another major case, *Brown Bag Software v Symantec Corp*36. In that case, the Court of Appeals for the Ninth Circuit advanced its own analysis which applies a technique called ‘analytic dissection’, although they did not articulate their test as clearly as did the *Altai* court. Because the *Brown Bag* test was developed as a response to the same criticisms of *Whelan* that *Altai* attempted to address, the two tests share many common features, particularly in their treatment of elements of expression that are unprotectable by copyright. The methodology is quite different, however.

The *Brown Bag* test is somewhat more difficult to grasp than *Altai*, being a creature of the case-law that developed in the US Ninth Circuit to analyse

---

This criticism of the merger doctrine cannot be levelled at the judgment in *Altai* because although Judge Walker refers to those traditional justifications for that doctrine, he explains the place of ‘merger’ reasoning in this stage of his analysis as resting on the lack of evidential value of similarities between works which are subject to similar constraints; 23 IPR 385 at 402, 403.

The merger doctrine can be supported on other bases; the landmark US Supreme Court decision in *Feist Publications Inc v Rural Telephone Service Co Inc* (1991) 20 IPR 129 provides two further explanations for the doctrine; first, it holds that copyright can only exist in original expression created by the author and not in underlying facts; ibid at 134. Although ‘ideas’ and ‘functionality’ were not specifically referred to by the court — the case involved a claim for copying of a telephone directory, so the court was concerned with a compilation of facts — the reasoning would apply with equal force to merger of expression with ideas and functionality. If there was only one or a very limited number of ways to communicate that fact, or idea, or to implement that function, then the author’s contribution could not be said to be ‘expressive’ and therefore copyright could not subsist. The other justification for the doctrine is the policy argument that the purpose of copyright is not only to protect the rights of authors but also to encourage others to build freely upon ideas or information contained in a work; ibid at 135. Where ‘merger’ occurs, then, policy dictates that copyright protection should not extend to that aspect of the work in order not to foreclose on the use of that merged expression by others.

---

36 960 F2d 1465 (9th Cir. 1992).
substantial similarity between copyright works generally. Perhaps for this reason, it has had practically no impact in non-US jurisdictions, unlike Altai. Within the US, however, it appears to be considered an equally important authority in this area even though, it is submitted, there is little to commend the Brown Bag test over Altai. Its significance in US cases dealing with this problem is partly to be explained by the fact that the Ninth Circuit includes the State of California which is one of the major centres of the computer industry. The large entertainment industry in California probably also contributes to the influence of the Ninth Circuit Court of Appeals decisions in the area of copyright law generally.

A reasonably comprehensible explanation of the Brown Bag test is to be found in one of the judgments in the Apple Computer Inc Microsoft Corp & Hewlett-Packard Co. series of litigation:

The Ninth Circuit employs a two-part test to determine whether a work infringes the copyright in another work. First, the ‘ideas’ of the works in suit are compared for substantial similarity, using an ‘extrinsic test’ or ‘objective analysis of expression’ ... Analytic dissection, employing a list of criteria of comparison informed by expert testimony, is a part of this exercise ...

If the ideas are substantially similar, then an ‘intrinsic test’ or ‘subjective analysis of expression’ is used ... [T]he ‘intrinsic test’ entails a comparison of the portions of a work that can be the subject of copyright protection ... Because the ‘intrinsic test’ is thus limited, a finding of similarity of ideas does not complete the ‘extrinsic test’ ... If the similarity of the works in suit stems solely from unprotectable features, then the plaintiff’s case is missing an essential element of infringement ...

The ‘non-literal components’ of a computer program, including its user interface, are protectable if, ‘on the particular facts of each case, the component in question qualifies as an expression of an idea [not] an idea itself’ ... To determine whether similarities result from unprotectable ideas, analytic dissection of similarities may be required. ‘If this demonstrates that all similarities in expression arise from use of common [ie, non-protectable] ideas, then no substantial similarity exists’ [citations and footnotes omitted]38

37 (1992) 24 IPR 225. The California District Court which heard this series of litigation was bound by the decision of the 9th Circuit Court of Appeals in Brown Bag.
38 ibid, at 234 to 235.
And this is as easy an explanation of the test as can be found! Part of the reason why the test is so complicated is that it combines the traditional substantial similarity analysis with guidelines as to when expert evidence should be admitted in that analysis. It is not helpful that there is no clear explanation of just what exactly is meant by ‘analytic dissection’ in this extract (or in the rest of the judgment, and *Brown Bag* itself), but that appears to be a reductionistic exercise of identifying ‘elements’ in the work and analysing them in isolation, instead of comparing the works as a whole.

The ‘extrinsic test’ concentrates on comparing the ideas in the work, on the assumption that if there is no substantial similarity between the ‘ideas’ embodied in the two works being compared, the works are not ‘substantially similar’ no matter how alike their respective expression. Once that is done, the ‘intrinsic test’ involves analysis of similarities in what has been identified to be protected ‘expression’ in the allegedly infringed program. This analysis combines to some extent the filtration and comparison stages in *Altai* although it begs the question of exactly what non-literal aspects of a program’s ‘expression’ are protectable in the first place.

The judgment in *Microsoft* goes on to explain that the process of analytic dissection (which occurs during both the ‘intrinsic’ and ‘extrinsic’ tests) applies doctrines for discarding unprotectable expression such as expression which is merged with idea, the *scene a faire* doctrine and functionality. These correspond with the considerations raised by the *Altai* court at the filtration stage.

The significant difference between the tests is that *Altai* is more thorough in requiring the entire program to be considered at all levels of abstraction, while the *Brown Bag* test appears to consider only two levels of abstraction — ‘idea’ and ‘expression’ — without indicating quite what these are. Then, at the ‘comparison’ stage, *Altai* requires the ‘substantiality’ of copied expression to be considered in relation to all the protectable expression in the allegedly infringed program while the ‘extrinsic’ analysis in *Brown Bag* involves considering only such similarities in protected ‘expression’ as are identified by expert witnesses.

The intellectual demands that these tests place upon users has led to understandable confusion. Terminology and ideas from both the *Altai* and *Brown Bag* tests have been used interchangeably in later cases without taking care to recognise the differences between the two tests. In at least one case, a court which was bound by neither decision described the *Altai* test in detail before stating that it would apply the *Brown Bag* test. Descriptions of the analysis then performed by the court used terminology drawn from *Altai* and even the discredited *Whelan*!

---

39 *ibid*, at 235 to 237.
40 *Atari Games Corporation v Nintendo of America* 25 IPR 646 at 653.
If one cuts through the detail, it is immediately clear that both these tests are far less favourable for copyright owners than the Whelan test. The Whelan test assumes that only the most general statement of the purpose of a program is unprotectable idea. It leads us to the conclusion that copyright may be infringed so long as the alleged infringer copied from the structure of the allegedly infringed program when some alternative implementation was possible, even if that alternative implementation was less desirable. The Altai and Brown Bag tests, on the other hand, call for dissection of the program, scrutiny of parts of the program in isolation and their elimination from the analysis if they prove not to be copyrightable for a number of reasons which were not considered in Whelan. What is left of the allegedly infringed program forms the basis for determining whether copying has taken place.

It is ultimately possible to say that the difference between the Whelan test and the two new tests is really a difference between the judicial philosophies at the heart of copyright; the former treats copyright as being principally to reward creativity and effort so maximum protection is desirable, while the later tests view copyright in computer programs as a purely utilitarian tool to maximise scientific and cultural advance, so no copyright should subsist where it is likely to impede progress. This article is not an appropriate forum to discuss the larger policy debate about the scope of copyright so criticism of Altai will be based on purely pragmatic considerations.

Criticism of Altai

Although these two tests are quite similar and serve the same purpose, this article will concentrate on criticism of the test in Altai since that test has proven influential outside the United States of America. Brown Bag need only be understood as the key to later US decisions that purport to apply that reasoning, but its strengths lie in areas already adequately covered by Altai — recognition of the characteristics of computer programs that affect the substantial similarity analysis — and any criticisms of Altai largely apply with equal or greater force to Brown Bag. At this point, it should be borne in mind that while the tests are directed towards determining 'substantial similarity', they effectively determine whether the subject matter of those similarities is copyrightable or not.

41 The judgments of courts in the United States do make frequent reference to these policy considerations because copyright law in the United States has a constitutional underpinning; see for instance the discussion in Gates Rubber Co. v Bando Chemical Industries, Ltd. 9 F3d 823 (10th Cir 1993) at 839 and Apple Computer v Microsoft, supra note 37 at 236.

42 It is tempting to observe that the similarities are a result of external constraints; both tests are intended to achieve the same purpose (sifting unprotectable idea from protectable expression) in relation to the same subject matter (computer programs), and both are influenced by the same criticisms of Whelan.
The basic weakness of these tests is their tendency to reduce the computer program under consideration to a number of basic features which are then individually considered for copyrightability: Altai does this expressly at the ‘abstraction’ and ‘filtration’ stages while in Brown Bag, this occurs at both ‘extrinsic’ and ‘intrinsic’ stages as ‘analytic dissection’. Since computer programs are functional works, almost every element of any program exists to contribute to that functionality rather than for any aesthetic reason. Taken in isolation, the elements making up a program contribute by performing their allotted sub-tasks and nothing more. With low-level elements (that is, relatively simple groups of code to perform very basic sub-tasks), there is a strong prospect that the ‘filtration’ stage will find most of such elements to be unprotectable in isolation; the code making up that element could fall into the public domain, or is likely to be very much constrained by the function to be performed and the dictates of efficiency. Most such basic elements would therefore not be protectable by copyright taken in isolation.

If an analogy may be permitted, the Altai test of breaking down the program into its layers of abstraction and filtering the individual elements of each layer in isolation would, at the lowest level of abstraction be akin to breaking a novel up into individual phrases, considering whether each phrase in isolation was copyrightable and discarding from the analysis those phrases that were not. The danger of this reductionistic analysis lies in the fact that many phrases are too simple in isolation to be copyrightable. It would often be the case that the expression of a phrase is so much constrained by the message to be communicated that the idea of the phrase would ‘merge’ with its expression. So, if you took a novel and discarded phrases like ‘The year was 1924’, ‘China was in chaos’, and ‘Much of it ... was ruled by warlords’ because they were not sufficiently ‘expressive’ or because their expression was too much dictated by the meaning to be conveyed, little would remain of the literal text of the novel. That cannot be the correct approach to analysing copyright in any literary work, including computer programs. It is submitted that copyright analysis should not be performed in this manner since elements which are uncopyrightable in isolation may still make up a large part of a copyrightable whole.

This danger is especially acute in the analysis of computer programs because programs are generally written so that the major program tasks are performed by discrete integers of code; these will be referred to hereafter as ‘modules’. These modules may themselves be compilations of discrete sub-modules which perform the sub-tasks that make up the module’s task, and the sub-modules composed of even more basic sub-sub-modules. Ultimately, the basic units making up the program may be so simple that most do not qualify as protectable expression, like individual phrases or words in a novel. But the compilation of sub-modules (the module) and the compilation of compilations of modules (the program) could yet be copyright works even though the basic constituents of those compilations
are not themselves ‘expressive’. If we test every basic element of a program — individual lines of text — in isolation and discard most of them from our analysis as being uncopyrightable on its own, there is a danger of failing to see that each of those lines of text was part of a copyrightable compilation in the first place.

The Altai test viewed as a whole would avoid this danger since the function of unprotectable elements is still considered at higher levels of abstraction, but it will be argued later that the complexity of the test and the works which must be analysed using it makes it all too easy to lose sight of the wood for the trees.\textsuperscript{43} If the test is not performed in its entirety then it is heavily biased against finding a computer program to be copyrightable.\textsuperscript{44}

An illustration of this danger can be seen in one of the many judgments in the Apple Computer, Inc. v Microsoft Corp & Hewlett-Packard Co.\textsuperscript{45} series of litigation. Although the court in this case viewed the claim as being one for copying of an audio-visual work and not copying of the textual structure of a computer program, the analysis performed was the ‘analytic dissection’ technique laid down in Brown Bag. It is therefore a suitable illustration of the reductionistic tendency of these tests.

For our purposes, the important aspect of the claim by Apple was that Hewlett-Packard’s ‘New Wave’ graphical user interface was a copy of its own ‘Macintosh’ interface. To determine whether there was infringement of Apple’s copyright, the Court performed analytic dissection on the Macintosh user interface to determine if the two were substantially similar. Rather than comparing the two interfaces as a whole, the court considered similarities between individual elements of the interfaces item by item,\textsuperscript{46} and concluded that most of those elements, viewed in isolation, were not subject to copyright protection. This was only to be expected since typical elements such as ‘icons’ in the graphical user interface are extremely simple works whose meaning is meant to be clear to even the untrained user. The likelihood of merger of idea with expression was thus extremely high for individual items.

Of the elements of Apple’s interface that remained after this analysis (which were very few), the court held that similarities with the defendant’s work

\textsuperscript{43} See the main text accompanying notes 47 to 49, supra.
\textsuperscript{44} One writer, commenting on this test, suggests that the end result of the Altai analysis is that only copying of the literal code or close paraphrasing is likely to amount to infringement; see D Karjala, ‘Recent US and International Developments in Software Protection’ (1994) 16 EIPR 58 at 60. Although the writer (and this discussion) does not refer to the other approach advocated in Brown Bag, a similar reductionistic tendency is likely to prevail there since it concentrates on comparing isolated similarities between works.
\textsuperscript{45} 4 CCH Computer Cases 46 860.
\textsuperscript{46} Ibid, at 65 247–9. In fact, the decision to adopt a reductionistic analysis to the ‘work’ was disposed of in an earlier interlocutory application in the District Court; see (1992) 24 IPR 225.
were not ‘substantial’ and so infringement was not established. The decision on the facts of the case was almost certainly the correct one, but it may be noted that the use of analytic dissection encouraged this reductionistic process which tends to lead triers of fact away from considering whether the combination of protectable and unprotectable elements of the user interface (or other work under consideration) amounts to a compilation. Even if the court had viewed the works as a whole and found a protectable compilation in the entire graphical user interface, other considerations may have been available to defeat Apple’s claim.

The ‘abstractions’ stage of the Altai test would appear to preempt this danger because it requires consideration of the program in question at all levels of abstraction. Protectable expression existing at higher levels of abstraction, such as a compilation of the constituent modules or sub-modules making up the program, will be subjected to ‘filtration’ separately from the basic elements and would be recognised as containing protectable expression under normal copyright principles. Ideally, higher level ‘expression’ would be saved for the ‘comparison’ exercise later even if all of the basic elements making up that higher level ‘expression’ were found not to be protectable when considered individually at the lower level of abstraction. It is submitted, however, that this supposed safeguard is illusory because the abstractions stage is impossible to implement in reality. The ‘abstractions’ stage requires the trier of fact to dissect the whole program, layer by layer, encompassing within his mind the elements of each of those layers and their intricate interrelationships within the layer and between different layers. With computer programs of any value being large and complex (at the lowest level of abstraction alone, perhaps consisting of hundreds of thousands of lines of code), no judge could possibly perform that task. None of the courts in cases which have adopted and purported to apply the test (including the trial court in Altai) have properly applied the ‘abstractions’ stage, very likely because it is impossible to do so. Instead, after paying lip service to this stage, they have (understandably) performed filtration and comparison on only apparent similarities between allegedly infringed and infringing programs which were identified by expert witnesses. In practice, then, the Altai analysis works very much like that suggested in Brown Bag.

Another criticism that may be levelled at the ‘abstractions’ stage of the Altai test is that it works on the assumption that ‘layers of abstraction’ exist in reality, and are distinct entities capable of being identified and considered in isolation from other ‘layers’, rather as one may separate the layers of an onion and study each separately. This is, however, clearly not

---

47 Even Judge Walker in Altai may have recognised this; see note 31, supra.
48 Infra, note 50.
49 775 F Supp 544 at 561 & 562. This reasoning was largely approved by the Court of Appeals for the 2nd Circuit.
the case with layers of abstraction in literary works. As abstractions, the ‘layers’ lack objective existence and their content and relationship with other layers will very much depend on the particular subjective perception of the viewer. In computer programs, the relationship of the behavioural structure to the other levels of structure is particularly difficult to visualise and does not fit comfortably into the scheme described by the court. The use of standard programming techniques may permit consistency among expert witnesses undertaking the task of identifying ‘layers of abstraction’ but this cannot be taken for granted.

Finally, one may ask if the Altai test has actually answered the central question bedeviling this area: at what level of abstraction does structure of a computer program become protectable by copyright? Performance of the analysis might reveal that a particular level of structure — or part of it — is not protectable because it is dictated by considerations of efficiency or for some other reason, but that is only likely to be true for extreme cases where it would have been reasonably clear that there are strong arguments against granting protection in any event. The test leaves a penumbra of doubt; intermediate levels of structure between literal text and highest, most general statement of function, may be capable of alternative implementations, but how many alternatives must there be for that level of structure to amount to ‘expression’? The analysis here is intuitive, and the Altai court, understandably, declined to attempt to spell out the analysis further.

In summary, the new tests are much more sophisticated and better account for the reality of computer technology than Whelan, but none of them has yet struck the right compromise between technical accuracy and practical usefulness. Altai identifies new factors to be weighed in deciding whether elements in a program taken in isolation should be protected but, it is submitted, the value of the test as a whole is outweighed by its reductionistic tendency.

**Developments After Altai**

Decisions claiming to apply the Second Circuit Court of Appeals’ decision in Altai have tended to adopt the test in that case uncritically\(^{50}\), though not

\(^{50}\) See for instance Sega Enterprises Ltd v Accolade, Inc., 977 F2d 1510, 1524-5 (9th Cir 1992); CMAX/Cleveland, Inc. dba Computermax v UCR Inc. 804 F Supp 337 (MD Ga 1992); and Autoskill, Inc. v National Educational support Systems, Inc. 793 F Supp 1557 (DNM 1992) and 994 F 2d 1476 (10th Cir 1993).

In Lotus Development Corporation v Borland International, Inc., the trial judge, Judge Robert Keeton, used a different methodology in disposing of an interlocutory application prior to the publication of the Court of Appeal judgment in Altai; 788 F Supp 78 (DC Mass 1992). But shortly after the Court of Appeal’s judgment in Altai, he re-considered the approach he had taken and concluded that it was ‘compatible substantively, though different in methodology’ from that of the Second Circuit Court of Appeals in Altai; 799 F Supp 203 (D Mass 1992), 211 to 212. Judge Keeton did not explain what he meant
always successfully. Perhaps an exception to this trend is the decision in *Gates Rubber Company v Bando Chemical Industries, Limited*. In their judgment, the US Court of Appeals for the Tenth Circuit not only explained the *Altai* test more comprehensibly but also contributed some useful comments about that test. Delivering the judgment on behalf of the court, Judge David Ebel noted that besides the three-part analysis, it would often be helpful to first look at the program as a whole to determine whether copying occurred. He also noted the relevance to this area of the leading US case on copyright in compilations, *Feist Publications, Inc. v Rural Telephone Service Co Inc* where the US Supreme Court affirmed that copyright protection could attach to an original organization, selection or arrangement of material even if copyright did not subsist in any of those items of that material making up the compilation. Awareness of these additional considerations could help to reduce the dangers of the reductionistic tendency inherent in the *Altai* test. Assuming that *Altai* remains the paradigm in the US, the additional points raised in *Gates* represent a step in the right direction towards an curing some of the shortcomings of that test.

**Conclusion: Altai**

It is too early to tell if the Altai test will come to dominate the jurisprudence in this area or whether it will be supplanted by a new paradigm. For the moment, the influence of that test is being felt in other jurisdictions such as the United Kingdom and in Canada. As the recent cases from England considering the *Altai* analysis will be discussed in the next section of this article, further discussion of the applicability of that test to UK or UK-based copyright law will be deferred for now.

---

51 When he said that the two approaches were ‘compatible substantively’ but it is likely that he felt that the tests would lead to the same result since they considered the same factors when attempting to distinguish copyrightable material from the uncopyrightable. It is perhaps instructive that the judge, who was not bound by the decision of the Second Circuit Court of Appeals, should have felt so constrained to explain how his decision fell in line with the new paradigm.

52 The attempt at application of the *Altai* test in *CMAX/Cleveland, Inc. dba Computermax v UCR Inc., ibid*; and *Autoskill, Inc. v National Educational support Systems, Inc., ibid*, was criticised in Karjala, ‘Recent US and International Developments in Software Protection’, supra, note 44 at 61.

53 supra, note 41.

54 (1991) 20 IPR 129.

55 Although developments in Canada are not being considered in detail in this paper, Canadian cases considering *Altai* are discussed in the main text accompanying notes 106 to 110. infra.

56 See the main text accompanying notes 82 to 105, infra.
(ii) User Interfaces

Some use has also been made of both the Altai and the Brown Bag tests to determine whether copying of a computer program’s user interface amounts to infringement of copyright in the program. In *Lotus v Borland*, the trial judge considered the Altai test, ‘filtered’ the structure and arrangement of menu commands of the Plaintiff’s program and found that they were protectable since there were many different ways in which these commands might possibly be implemented. In the *Apple v Microsoft* series of litigation, Judge Vaughn Walker applied the Brown Bag test to a claim of copying of a ‘graphical user interface’. He found that the overall look and feel of the plaintiff’s graphical user interface was purely functional or utilitarian, and so was not protectable by copyright. He went on to analyse individual items of similarity between the two interfaces and found likewise.

Although the facts of the two cases are very different, and there were significant differences in the merits of the respective plaintiffs’ cases, a similar legal issue was involved in each case and resolved differently by the two judges. The differing results may in part be a result of the differing methodology used. It is submitted that the reductionistic tendency in the Brown Bag test inclined Judge Walker in *Apple* to dismiss the graphical user interface as a utilitarian article because of his focus on the individual elements of that interface and the functionality of each of those elements. In the process, he tended to overlook their expressive quality. In contrast, Judge Keeton in *Lotus* was prepared to look at the entire menu structure as a whole and, viewing it from that high level of abstraction, he found a compilation of individual commands and groups of commands. This approach has been criticised for its failure to adhere to copyright tradition in the United States as exemplified by the US Supreme Court decision in *Baker v Selden*. That ‘flaw’ in its reasoning might be a strength if we were to consider the applicability of that decision in Singapore and other Commonwealth jurisdictions, since it has been suggested that an English court might have decided *Baker v Selden* differently.

The different results in *Apple* and *Lotus* can also be explained if we note that the evolution of user interfaces may call for different approaches to copyright protection. *Lotus* represented the first generation of user interface cases; the claim here was essentially that the Plaintiff had invested effort...
and resources to teach users to operate the program and thus built up a base of customers. Lotus’ goodwill lay partly in the fact that once users were familiar with its program, it would be difficult for a rival program to compete for their loyalty because of the costs and inconvenience the customers would face in learning to operate the rival program. The feature in the defendant’s program which led to the claim allowed it to ‘emulate’ the Lotus user interface — the user of the defendant’s program could choose to operate that program using the same commands as the plaintiff’s program. The case may thus be viewed as one involving ‘unfair appropriation’ of the goodwill that Lotus had invested heavily to build up.

In contrast, Apple in its action was seeking to protect its user interface not because it had invested in training a pool of users, but because it had invested in developing a system whose value lay in its ease of use. Users who were familiar with the plaintiff’s graphical user interface could operate a large number of applications programs with little difficulty. This strategy led Apple to produce a user interface which would be intuitively appealing to the user. The different commercial considerations in Apple highlighted a different policy consideration: the concern that conferring strong copyright protection on this user interface would prevent rival programmers from acting upon the commercial imperative to improve their programs by introducing user interfaces having similar appeal.

(iii) Reverse Engineering

The practice of ‘reverse engineering’ of computer programs gives rise to problems in copyright law that would not occur for other literary works. Reverse engineering typically involves ‘decompiling’ a computer program from the machine-readable form in which most programs are released to the public (called ‘object code’) into a form which is capable of being understood by human programmers (‘source code’), and the process of studying and analysing the resultant source code in order to understand how that program was designed and how it behaves when run. Although the process approximates the translation of a normal literary text from one language to another, the process of ‘decompilation’ is considerably more complex. In any event, decompilation results in the reproduction or adaptation of the decompiled program so the question arises whether and to what extent decompilation of computer programs ought to be permitted by copyright law.

There have been two recent US decisions of significance dealing with reverse engineering of computer programs; Sega Enterprises Ltd. v Accolade, Inc.62 and Atari Games Corporation v Nintendo of America Inc.63. Both cases

62 977 F 2d 1510 (9th Cir 1992).
63 Supra, note 40.
involved claims by manufacturers of video game hardware and software that their games programs had been infringed by rival manufacturers who had ‘reverse engineered’ their programs. The source code thus derived was used by the defendants to produce their own games software that could run on the plaintiffs’ hardware notwithstanding hardware and software devices in the plaintiff’s products which were meant to prevent this. The ‘reverse-engineering’ was found in *Sega* to fall within the statutory ‘fair use’ defence, while the defendant in *Atari* was found to have infringed.

The decision in *Sega* and its implications for the law of Singapore are dealt with in a recent article in this journal\(^{64}\), and comments made there on the fair use defence apply with equal force to the *Atari* decision, so little more need be said about either case here except to note that the analysis of the court in *Atari* may be criticised as being seriously deficient in attempting to apply the leading US authorities on computer program copyright\(^{65}\). Also, it should be noted that there is a significant difference between the facts of *Sega* and *Atari* that probably contributed to the difference in result. While the defendant in the former case obtained the source code by decompilation of the plaintiff’s program only, the defendant in the latter case improperly obtained a copy of the plaintiff’s source code when decompilation failed to provide the necessary information, and copied more of this code than was strictly necessary to overcome the locking feature in the plaintiff’s products. The latter was thus not a straightforward case of reverse engineering, and the defendant’s improper conduct in that case had a significant impact on the outcome.

(iv) *Feist Publications v Rural Telephone Services Co*

There is one last development in the US that needs mention before we move on to the next section considering developments in the UK. No survey touching on copyright law developments in the US would be complete without directing readers’ attention to the very important US Supreme Court decision in *Feist Publications v Rural Telephone Services Co*\(^{66}\). Although that decision is not truly ‘recent’ as that term is used in this article, and it was not concerned with computer programs (it related to a claim of infringement of copyright in a factual compilation) the judgment in that case has profound implications for US copyright law generally.

Perhaps the most significant holding in that case is the court’s rejection of the ‘sweat-of-the-brow’ theory of copyright protection for factual compilations. The court instead asserted that effort expended in compiling facts alone is not sufficient to create a compilation in which copyright subsists.

---

\(^{64}\) Supra, note 29.

\(^{65}\) See the main text accompanying note 40, supra.

\(^{66}\) Supra, note 54.
and that some creativity must be exercised for a compilation to obtain copyright protection. This view may be compared with UK decisions such as *Ladbroke (Football) Ltd v William Hill (Football) Ltd* and *Waterlow Directories Ltd v Reed Information Services Limited* to shed light on differences between the basic principles of copyright protection under US and Commonwealth copyright law; UK courts seem much more prepared to accept that effort alone without any creativity in selection or arrangement can nonetheless result in a copyrightable compilation. *Feist* may prove to be the point of irreconcilable divergence between the systems of copyright law, or it may yet lead to a convergence if we can clarify the basic assumptions and terminology used in UK and other Commonwealth cases.

That decision is particularly important to the question of copyright in computer programs because claims for infringement of computer programs will very often relate to complex works that are better viewed as a compilation of computer programs. The decision in *Feist* (or its rejection by a UK or local court) would thus be relevant to the question of whether and to what extent copyright subsists in such a work.

### III. RECENT DEVELOPMENTS: THE UNITED KINGDOM

#### a. Copyright Law

*(i) The UK Copyright (Computer Programs) Regulations 1992*

An important development to be noted here is the amendment of the UK Copyright, Designs and Patents Act 1988 as it applies to computer programs. The amendments came into force on the 1st of January 1993 and were effected by the UK Copyright (Computer Programs) Regulations 1992 in response to the EC Council Directive on the Legal Protection of Computer Programs. That directive was intended to lead to the harmonisation of the laws of Member States pertaining to intellectual property rights in computer programs.

The basic principles of copyright law remain largely untouched by the Regulations and changes introduced are (with one exception) specific and relatively minor refinements to the rules of copyright as they apply to

---

69 This question of computer programs as compilations is considered in the main text accompanying note 43, *supra* and note 97 and 98, *infra*.
70 1988, c 48; hereafter ‘the CDPA’.
71 SI No.1992/3233.
computer programs. Only brief mention need be made of the following:

(1) copyright is now expressed to subsist in ‘preparatory design material for a computer program’;\(^ {73}\)

(2) the doctrine of ‘exhaustion of rights’ in computer program copyright is given specific statutory expression;\(^ {74}\)

(3) a specific definition of ‘adaptation’ of a computer program has been introduced; it now means ‘an arrangement or altered version of the program or a translation’ of the program;\(^ {75}\)

(4) acts incidental to the normal use of programs such as making of back up copies or maintenance are specified not to amount to infringement of copyright;\(^ {76}\)

(5) the class of persons against whom a copyright owner may bring infringement action for dealing in devices designed to circumvent copy-protection of computer programs has been extended to persons who possess such devices in the course of business;\(^ {77}\) and

(6) restrictions placed by copyright holders upon the making of back-up copies, decompilation for the purpose of producing interoperable programs and studying and testing programs may be void.\(^ {78}\)

The major change introduced by these regulations is the enactment of specific provisions creating a very limited right to reverse-engineer computer programs.\(^ {79}\) Such conduct can no longer come within the ‘fair dealing’ provisions of the UK Act.\(^ {80}\) The new Section 50B of the Act provides that it is not infringement for a lawful user to decompile a program subject to the following conditions:

(a) the decompilation is necessary in order to obtain “information necessary to create an independent program which can be operated with the decompiled program or with another program (‘the permitted objective’)”;\(^ {81}\)

(b) information thus obtained is not used for any other purpose than the permitted objective;\(^ {82}\)

---

\(^{73}\) *Supra*, note 71, regulation 3, which amends the definition of ‘literary work’ in section 3 of the CDPA.

\(^{74}\) *Ibid*, regulations 4 and 6, amending sections 18 and 27 of the CDPA respectively.

\(^{75}\) *Ibid*, regulation 5, amending section 21 of the CDPA.

\(^{76}\) *Ibid*, regulation 8, inserting new sections 50A and 50C.

\(^{77}\) *Ibid*, regulation 10, amending section 296 of the CDPA.

\(^{78}\) *Ibid*, regulation 11, inserting a new section 296A.

\(^{79}\) *Ibid*, regulation 8, inserting new sections 50A, 50B and 50C.

\(^{80}\) *Ibid*, regulation 7, amending section 29 of the CDPA.
(c) the reverse engineer does not have ‘readily available to him’ the information necessary to achieve the permitted objective;

(d) decompilation is restricted to only such acts as are necessary to achieve the permitted objective;

(e) the resultant information is supplied only to persons for whom such information is necessary to achieve the permitted objective; and

(f) the reverse engineer uses the information only to create programs which are not ‘substantially similar in [their] expression to the program decompiled’.

The interpretation of these Regulations and detailed consideration of the problems which they raise fall outside the scope of this article and are considered elsewhere 81. It is only necessary to point out that the new rules in England on copyright protection of computer software mean that decisions from the UK on reverse engineering, if they should ever occur, will be even less relevant to the law of Singapore than at present. The fact that most other basic copyright principles are untouched, on the other hand, permits us to continue to look to English cases as persuasive authority on such matters as the approach to dealing with non-literal copying.

(ii) Cases considering Altai

To date, two English cases have considered the issue of non-literal copying of computer programs in the light of the Altai methodology. The first, John Richardson Computers Limited v Flanders 82 was favourably disposed towards that methodology, while the more recent case, Ibcos Computers Ltd v Barclays Mercantile Highland Finance Ltd 83 was sceptical of the value of American authorities in this area.


82 [1993] FSR 497. Readers may also wish to have references to articles explaining and commenting on this judgment such as D Horan, “John Richardson Computers Limited v Flanders and another — a commentary” (1993) 9 CL&P 70; R Arnold, ‘Infringement of Copyright in Computer Software by Non-textual Copying: First Decision at Trial by an English Court’, (1993) 15 EIPR 250; and Ng-Loy Wee Loon, ‘Reverse Engineering of Computer Programs — Rethinking its Prohibition’, supra, note 29.

83 Supra, note 61.
John Richardson v Flanders

For our purposes, the important facts are that the plaintiff had employed the defendant to work on the plaintiff’s program which performed labelling and stock control in chemist’s shops. They parted company and shortly thereafter, the defendant produced a rival program which was the subject of the copyright infringement claim. The literal code of the two programs was not similar and the defendant’s program could not be said to have been a translation of the plaintiff’s since it was written in different programming language for a different type of computer. The plaintiff nevertheless argued that the defendant had copied the non-literal features of his program — the ‘general scheme of [the plaintiff’s] program, including the detail of certain routines of an idiosyncratic nature’.

The judge, Ferris J, approached the issue of copyright infringement by asking the following questions:

1. Were there objective similarities between the two programs?
2. If there were similarities, were they the result of copying?
3. If there was copying, was it of a ‘substantial part’ of the plaintiff’s program?

This is nothing more than a version of the logical analysis for determining infringement by copying where the fact of copying cannot be directly proved; one looks at whether the alleged infringer had access to the allegedly infringed work and whether there are similarities that are probative of copying. It was clear on the facts of this case that even if the defendant had not had ‘access’ to the plaintiff’s program in the sense of having reference to a copy of the plaintiff’s code as he wrote the program, the defendant had written several versions of that program himself and his familiarity with it was such that, at the least, he could have made use of his recollection of the plaintiff’s program in a manner that amounted to reproduction. Something akin to ‘access’ having been established, the remaining analysis involved asking those first two questions identified by the judge to determine whether there were objective similarities between the works which were probative of copying.

The last question — whether there was copying of a ‘substantial part’ of the plaintiff’s program — is mandated by Section 16(3) of the UK Copyrights, Designs and Patents Act 1988 which expressly provides that references to a work (for instance, in provisions on infringement by copying the work) apply also to a ‘substantial part’ of it. Whether a part of the
work is ‘substantial’ depends on the quality of what is taken rather than its quantity.

After reviewing the evidence, Ferris J felt all these questions were answered in the plaintiff’s favour and infringement was found. The part of the judgment dealing with these issues is almost entirely devoted to factual analysis of the evidence. This is curious since in an earlier part of his judgment the judge considered in some detail legal arguments made by counsel for the parties and in particular commented upon the applicability of decisions from the US; but when the time came in the judgment to apply his legal conclusions to the evidence, he did not do so clearly.

It is the earlier legal reasoning that is of interest to us, though the process of critically analysing this part of the judgment is complicated by the fact that for most of this discussion, the judge simply recounted arguments made by counsel without comment. The only legal propositions which we can be sure the judge accepted himself are his important dicta on the proper use of the American authorities, and his warning to himself (based upon an earlier English decision, Thrustcode Ltd v W.W. Computing Ltd) that similarities between screen displays of computers are to be treated with caution in infringement analysis.

It was in the context of an examination of the case law on claims for copyright infringement of computer programs that Ferris J considered the judgments in Whelan and Altai. He briefly described the two approaches and, ignoring Whelan thereafter, commented:

There is nothing in any English decision with conflicts with the general approach adopted in the Computer Associates case. I think that in preference to seeking the ‘core of protectable expression’ in the plaintiff’s program an English court will first decide whether the plaintiff’s program as a whole is entitled to copyright and then decide whether any similarity attributable to copying which is to be found in the defendant’s program amounts to copying of a substantial part of the plaintiff’s program ... But at the stage at which the substantiality of any copying falls to be assessed in an English case the question which has to be answered, in relation to the originality of the plaintiff’s program and the separation of an idea from its expression, is essentially the same question as the United States court was addressing in Computer Associates. In my judgment it would be right to adopt a similar approach in England ... Nevertheless I find difficulty in applying the abstractions test to the particular facts of this case ...

86 Supra, note 82 at 528 to 558.
88 Supra, note 82 at 526–7.
The last statement of the quoted passage lends some support to the point made earlier\textsuperscript{89} that the Altai test (or at least the abstractions stage) is perhaps not truly usable in practice. If one goes on to examine the part of the judgment where Ferris J dealt with the question of whether the copying which was found to have taken place was of a substantial part of the plaintiff’s work\textsuperscript{90}, the Altai approach is not much in evidence. Instead, he stated:

‘I also consider that the filtration process described in the Computer Associates case will be relevant in assessing substantiality, although I admit to having considerable difficulty in knowing how some aspects of that process are intended to work’.\textsuperscript{91}

It is perhaps just as well that the Altai test is not a practically useful test because the apparent affinity between the terminology used in judgments in UK and US authorities only masks what, it is submitted, are significant differences in the methodology of infringement analysis between the two jurisdictions. As stated earlier\textsuperscript{92}, the critical, and often sole issue in US copyright infringement cases tends to be the determination of whether the works in question are ‘substantially similar’. This analysis actually combines the determination of whether there are similarities between the works which are probative of use of the allegedly infringed work by the alleged infringer together with the distinct questions of whether that use amounts to ‘reproduction’ of the allegedly infringed work, and whether the amount of the alleged infringed work which was used is so significant as to amount to infringement\textsuperscript{93}. The ‘substantial part of the plaintiff’s work’ analysis in

\textsuperscript{89} See the main text accompanying notes 47 to 49, supra.
\textsuperscript{90} Supra, note 82 at 548 to 558.
\textsuperscript{91} ibid, at 549.
\textsuperscript{92} See the main text accompanying notes 23 and 24.
\textsuperscript{93} See, for instance, the statement of the basic purpose for seeking out ‘substantial similarity’ in the Court of Appeal judgment in Whelan, supra, note 25 at 13–14. It was noted there that if the court is satisfied of substantial similarity between the programs and that the defendant had access to the plaintiff’s program, copying is proved. Clearly the test determines whether similarity between the works is probative of copying.

As the substantial similarity exercise is explained in Altai, the determination of substantial similarity is partly directed at ensuring similarities identified are probative of copying; \textit{supra}, note 35. But some of the factors weighed in the Altai substantial similarity analysis do not assist the assessment of the probative value of similarities and are instead relevant to the question of whether copying particular elements of a computer program can ever amount to copyright infringement --- for instance, the \textit{scenes a faire} doctrine, and the question of whether copied expression lay in the public domain. The true purpose behind these considerations is obviously to determine whether such a use of a copyright work can amount to ‘copying’ or ‘reproduction’, and whether the taking is sufficiently significant to amount to infringement. The US ‘substantial similarity’ analysis thus does not distinguish between these three logically distinct objectives and attempts to address them all at once.

A similar tendency to combine the analysis of whether similarities are probative of copying and whether similarities are sufficiently significant to amount to infringement is to be seen in the approach of the Court of Appeals for the 9th Circuit in Brown Bag, \textit{supra} note 26 at 1475–6.
UK cases is directed at only the last of these questions and, logically, should be performed only after copying has been established. The Altai ‘substantial similarity’ test is thus incompatible with what appears to be its equivalent in the traditional UK analysis.

Also, the reductionistic tendency inherent in the US methodology may be acceptable in the context of the multiple objectives served by the US ‘substantial similarity’ analysis, but there can be no justification for stepping away from the traditional approach under UK law of considering a copyright work as a whole.

A final criticism of the judge’s embrace of the Altai test is that he failed to recognise that the type of claim before him was significantly different from that in Altai. It is correct to say that the issue before him related to copying of non-literal aspects of the computer program, but the similarities that he identified between the programs were essentially similarities between the user interfaces of the two works. This is significant because the external constraints which condition the ‘expression’ (if any) in a user interface are not the same as those which affect the textual and other behavioural structure of the programs. The Altai approach has not actually been used to analyse similarities between user interfaces of programs94, and it is this writer’s opinion that the Altai approach requires considerable refinement before it can be used in that context95.

Ibcos Computers v Barclays Mercantile Highland Finance

Although the facts of this case are closer to Altai than those in John Richardson Computers, the judge in Ibcos, Jacob J, disagreed with Ferris J and took the opposite view of the applicability of the American approach. Again, the plaintiff and defendant were jointly involved in developing a software package called ADS, which was a system to computerise some of the administrative operations of agricultural machinery dealerships. The defendant eventually left the plaintiff’s employ and developed a rival software package, Unicorn. The important part of the plaintiff’s claim is its assertion that Unicorn was a result of copying ADS. The judge upheld the plaintiff’s claim of copyright infringement. As a landmark case in this area, there are already casenotes discussing the reasoning used96, and more will

---

94 It was pointed out in note 50 that the trial judge in Lotus Development Corporation v Borland International, Inc. felt constrained to show that his analysis of copying of a user interface was not incompatible with the Altai approach, but he did not attempt to apply that test at first instance, and his subsequent consideration of the application of that test was severely criticised; see note 44 at 65.

95 The US Court of Appeals for the Second Circuit also expressly cautioned that the test propounded there was not necessarily meant to apply to screen displays, which comprise part of the user interface; supra, note 22 at 396.

undoubtedly be written before long. This article will not attempt a detailed analysis of the judge’s reasoning and will consider only points of particular interest in understanding how copyright law should apply to computer programs.

Addressing the question of what aspects of the program were protected by copyright law, Jacob J noted that the package of programs as a whole was a compilation and protectable as such by copyright. It is submitted that the judge did not go as far as might have been desirable. He could have found the package to have been a compilation not only of a number of files (individual programs and data files that make up the software package) but also of modules within each of those programs. From those premises, the structure of the package as a whole and of its constituent programs could have been analysed in the context of copyright in the compilation; the ‘selection and arrangement’ of the modules and sub-modules. While it must be conceded that the fit is not a perfect one — ‘structure’ of a program is more complex than the ‘selection and arrangement’ of items in a compilation of traditional literary works — it is submitted that the concept of copyright in a compilation has greater potential to develop and cope with the special problems raised by non-literal aspects of computer programs than the general principles developed for plots of novels. Copyright in a compilation is not tied to the copyrightability of the underlying text as is copyright protection of non-literal aspects of literary works such as the plot of a novel. It is also a less intuitively accessible concept — how does one visualise the ‘selection and arrangement’ of works within a compilation — and is thus less likely to be decided on the basis of inaccurate analogies with existing copyright works. If we must protect computer programs with copyright, then the law on copyright in a compilation is perhaps less inappropriate for the protection of program structure than copyright in traditional literary works. Viewed in that light, this finding by Jacob J represents a good starting point, and it is to be hoped that this line of analysis develops further to a logical conclusion.

As in the John Richardson Computers case, the major point of interest for us is the judge’s attitude towards US authorities on copyright in computer programs. Jacob J considered their applicability in relation to the question of whether copyright subsisted in the work, and whether copying which he found to have taken place was of a ‘substantial part’ of the work. On both of these questions, he declined the guidance offered by the US authorities,

97 Supra, note 61 at 289.
98 The Singapore Copyright Act (Cap 63, 1988 Rev Ed) facilitates such an approach to copyright in structure of computer programs because the definition of the term ‘literary work’ in section 7 expressly includes “a computer program or compilation of computer programs” [emphasis mine]. Provided the constituent modules making up a program under consideration fell within the definition of ‘computer program’ in Section 7 themselves, the ‘structure’ of that program would in part comprise the compilation (ie the arrangement and selection) of modules.
citing differences between the copyright laws of the US and UK. He observed that US copyright law, unlike UK law, contained an express statutory exclusion of copyright protection to functional works. As such, he felt that the US authorities on subsistence of copyright in computer programs would not assist an English court since they were founded on a statutory provision which did not appear in UK copyright legislation. The correctness of this proposition cannot be taken for granted; UK copyright legislation certainly does not expressly exclude copyright protection for ‘ideas’ or ‘function’, but case law in the UK has recognised limits on the subsistence of copyright corresponding closely to the subject matter excluded by statute in the US so this difference between US and UK law may be more apparent than real. Determination of whether there is in fact such a wide substantive difference between the two legal systems as justifies the judge’s conclusion lies beyond the scope of this survey, but even if Jacob J was incorrect on that point, it is submitted that he was still partly correct to reject the appeal to US authorities. The significant difference between the methodology used in US and UK courts to determine copyright infringement makes it inappropriate to graft the US analysis onto UK (and by extension, Singapore) law.

Another aspect of these decisions worth noting is that they nicely illustrate the differences in methodology between the UK and US approaches to the analysis of copyright infringement. Both judges in Ibcos and John Richardson devoted much of their judgments to consideration of whether similarities between the works was probative of copying, and the forensic analysis cannot be faulted. Jacob J in the former case devoted much of the introduction to his judgment describing facts which were important for that later analysis of whether copying had taken place. But once it was found that use had been made of the allegedly infringed work, the judges performed an impressionistic (and not fully explained) evaluation of whether the use was of a ‘substantial’ part of the allegedly infringed work. Jacob J in Ibcos expressly stated that the major consideration was whether there had been ‘overborrowing of the skill, labour and judgment which went into the copyright work’. Ferris J in John Richardson was no more explicit; ‘substantiality’ of copying was often determined by asking whether much time, skill and/or effort had been applied by the plaintiff to the creation of the feature which was found to have been copied. Neither judge was much detained by the special concerns raised in Altai about copyrightability.

99 Supra, note 61 at 292. The relevant provision of the US Copyright Act is set out in note 100, supra.


101 See the main text accompanying notes 92 and 93, supra.

102 Supra, note 61 at 302.

103 Supra, note 82 at 548 to 558.
of computer programs. The American judges, particularly those in Altai and Brown Bag, have on the other hand been much more concerned with the high metaphysics of copyright law and technical accuracy to the point that the tests they propound are of limited use in practice, while the forensic analysis of copying is diluted by the other aims of their ‘substantial similarity’ analysis.

Even so, it is submitted that Jacob J was too quick to reject the US authorities as unhelpful. Substantive and methodological differences between the US and UK decisions notwithstanding, there is much to learn from US case law, in particular, from their lucid and technically sophisticated discussions of computer technology. The first stages of the Altai test teach a great deal even if their practical value is limited; they cannot be applied directly but they may still inform our analysis. The ‘abstractions’ stage thus warns us that computer programs are extremely complex works; the trier of fact must appreciate that complexity and the relationships between levels of structure within the work, and care must be taken not to lose sight of the higher levels of abstraction by focusing only on literal code. The ‘filtration’ stage identifies factors which are relevant in determining whether copying has taken place, as well as the significance of any copying that is eventually found.

One of the alleged acts of infringement in Ibcos which should have merited particular attention relates to what are called file transfer programs in the defendant’s program, Unicorn. The purpose of these file transfer programs was to convert data stored using the Plaintiff’s program, ADS, into a form directly usable by Unicorn. Such a facility substantially improved Unicorn’s attraction for existing users of ADS since they could immediately use their existing data files on Unicorn without going to the trouble of re-entering it using the Unicorn format. To write these file transfer programs, the Defendant had to use the file layouts of ADS’ data fields. The judge found as a fact (and without detailed consideration) that these had been copied and that the layouts constituted a ‘substantial part’ of the Plaintiff’s program. The lack of discussion on this point is to be regretted since the part of the claim gives rise to the interesting question of whether elements of this nature can in fact be protected by copyright. Had the judge considered the factors that lie behind the US doctrine of ‘merger’, he might have come to a different conclusion about protecting such a feature. Such reasoning would have had significant implications for later user interface claims as well as the broader policy question of the kind of competitive behaviour that copyright permits, but this was not to be. It is perhaps consistent with the usual approach of English judges to avoid

104 Although he did explain earlier in his judgment that the definition of the variables which make up the layouts is ‘part of the creative process’; supra, note 61 at 287.
105 The doctrine of merger is discussed in note 35, supra.
detailed, policy-based reasoning in relation to such an overtly policy-based question, and is perhaps understandable in this case where this was only one of many issues that the court had to grapple with, but one cannot help feeling that an opportunity to provide some direction for this area of the law was lost.

A Brief Look at Canada

Having argued against too quick acceptance of the Altai test, it should be pointed out before leaving this part of the article that a report in a recent issue of the European Intellectual Property Review states that courts in Canada, another Commonwealth jurisdiction, have been less reticent about adopting the Altai test\textsuperscript{106}. Although Canada does not fall within this survey proper, that development stands in stark contrast with the decision in Ibcos, and is of particular interest since Canadian copyright rules are similar to those of the UK.

The adoption of Altai is reported to have happened in a Quebec Superior Court case, Matrox Electronic Systems v Gaudreau CS Montreal\textsuperscript{107}. The detailed judgment in that case is not available at time of writing so detailed comment is impossible. That note in the Review makes reference to another Canadian case which supposedly adopted the Altai approach, Delrina Corp. v Triolet Systems, Inc.\textsuperscript{108} however a review of the judgment in that decision shows that it may have been putting it too strongly to say that the court there ‘followed’ the Altai approach. Altai was cited in that case in relation to the question of whether the user interface of a computer program could be protected by copyright and was only one of several decisions cited on that point, together with other significant US cases such as Lotus v Borland\textsuperscript{109} and Apple v Microsoft\textsuperscript{110}. Although all these cases adopted different approaches towards the analysis of substantial similarity, the Canadian judge made no reference to this, nor expressly preferred any one approach. Instead, he drew from them the conclusion that computer programs necessarily contained elements which could not be protected by copyright for the reasons mentioned in the US cases. A necessary step in the analysis of copyright infringement was therefore the use of some method for weeding out those elements. This conclusion, and the analysis of facts which followed, is equally consistent with the methodology adopted in UK decisions and cannot accurately be described as adoption of the Altai approach.

\textsuperscript{107} CA Montreal 500–09–001580–935
\textsuperscript{108} [1993] 47 CPR (3d) 1; 1993 Ont.CJ LEXIS 219.
\textsuperscript{109} Supra, note 57.
\textsuperscript{110} Supra, note 46.
b. Patent Law

There are two recent UK cases on patent protection of computer-related inventions to be noted; *Lux Traffic Controls Limited v Pike Signals Limited* and *Raytheon Co’s Application*. These decisions have neither changed nor clarified the law significantly, but merely re-state some basic rules laid down in earlier cases such as *Merrill Lynch’s Application*, *Re Gale’s Application* and *Hitachi Ltd’s Application*. Those basic rules are quite clear and can be stated in a number of simple propositions:

1. An invention is not unpatentable simply because it involves the use of a computer and/or a computer program.

2. A claim reciting nothing more than a computer program is unpatentable. Similarly, a claim which does nothing more than use a computer to implement a mental act, or a method for doing business is unpatentable.

3. The analysis of whether a claim recites nothing more than a computer program (or a mental act, or method for doing business) ‘as such’ involves looking at the claim as a whole. It is the wrong approach to isolate the inventive step and determine whether that consists only of a computer program.

Where case law presently falls short is in indicating clearly what we must look for when we consider the claim as a whole to distinguish unpatentable claims (those for computer programs ‘as such’) from patentable claims. Most of the cases turn on an examination of whether the claim makes a technical contribution to the art, or whether the invention provides a technical result. These references to ‘technical’ nature which distinguish a patentable invention from a computer program ‘as such’ are a result of consideration by UK courts of decisions made by the European Patent Office under the European Patent Convention, to which the UK is a

---

116 This follows from the words of section 1(2)(c) of the UK Patents Act 1977, that a computer program ‘as such’ will not be patentable; see main text accompanying note 14, *supra*.
117 Like Clause 13(2) of our Patents Bill, *ibid*, the UK Patents Act 1977 makes specific provision that a rule or method for performing a mental act, playing a game or doing business is also unpatentable. *Merrill Lynch’s Application, supra*, note 113, dealt with what was ultimately found to be a method for doing business.
118 *Merrill Lynch’s Application, supra* note 113.
119 *Ibid* at 569.
120 *Re Hitachi Ltd’s Application, supra*, note 115 at 422.
signatory. The UK Patents Act itself makes no reference to concept of ‘technical contribution’ or ‘technical result’ so UK courts have tended to avoid using this concept too openly\footnote{Some judges have gone further to expressly reject the concept of a technical effect as the touchstone of patentability in this area. In \textit{Wang Laboratories Inc.'s Application}, Aldous J expressed some scepticism of reasoning based on ‘technical’ nature of an invention; [1991] RPC 463 at 470 and 472. He stated that the term ‘technical’ was not sufficiently certain in English law to be of use.}. A requirement in UK patent law for ‘technical contribution’ as a condition to patentability would provide a consistent explanation for the results of the UK patent cases to date and allow it to develop in tandem with the law on European patents, but the approach of the courts makes it unlikely that a clear definition will soon emerge of what exactly ‘technical’ nature means in English law. It is likely that UK patent cases will continue to vacillate between attempts to apply the literal terms of the Patents Act 1977 and use of the concept of ‘technical’ effect, without clarification of either. The two latest cases fall within this trend.

The \textit{Lux Traffic Controls Limited}\footnote{\textit{Ibid}, at 139 and 143. In spite of his earlier dismissal of the test of ‘technical effect’, \textit{supra}, note 120, Aldous J appears to have reconciled himself to the use of that concept of ‘technical contribution’ in this case, even though it is no more certain what ‘technical contribution’ meant here than in the European decisions which caused his difficulty earlier.} case involved a computerised traffic light control system which used motion detectors to optimize traffic flow through a junction. Aldous J in the Patents Court accepted that if the claim had been confined to a system for controlling traffic without more (ie nothing more than a procedure for adjusting the duration and sequence of lights) it would not have been patentable; that claim would have been one for performing a mental act ‘as such’. The plaintiff’s claim went further, however, incorporating a method for accommodating the possibility of vehicles stalling at the junction through the use of motion detectors. These devices would determine if a vehicle at the junction did not move after lights turned green in its favour and the system would automatically prolong the green light to avoid a build-up of traffic in that direction. This was found to amount to a ‘technical contribution’\footnote{\textit{Supra}, note 112.} and so the patent was held to be valid. The judge did not attempt to explain in clear terms what a ‘technical contribution’ involved.

More recently, the Patents Court considered \textit{Raytheon Co’s Application}\footnote{\textit{Supra}, note 111.}. This case involved a claim for a system and apparatus for identification of objects such as ships using their silhouettes. This system had the ability to cope with silhouettes which were not properly registered or in non-standard sizes by converting the digitized image of the ship to be identified into a...
set of vectors which could then be processed and compared with a reference library of silhouettes of known ships until the best possible match was found. The method claim set out the series of steps, while the apparatus claim involved an image-forming device, a device to digitize the image thus formed and a computer to do the comparison. The examiner in the Patent Office objected to the method claim on the grounds that this was no more than a method for performing a mental act, while the apparatus claim was simply a conventional computer with a program which performed a mental act.

The Court upheld the objection, ruling that the claims achieved no technical advantage or technical contribution. They therefore related respectively to a mental act as such and a computer program as such and were thus unpatentable. Although the judge’s factual analysis is not expressed in the judgment, it is submitted that the finding that the claims lacked the requisite ‘technical’ character was a result of the fact that the output produced by the claimed apparatus was nothing more than abstract information — the likely identity of the ship in question. It could have made all the difference if the claimed apparatus had an additional feature which used the output information to produce some physical or tangible result — what is referred to in some US cases as ‘post-solution activity’.

The judge cited and distinguished the well known decision of the European Patent Office Technical Board of Appeal in VICOM Systems Inc./Computer-related invention, which involved a successful claim for a patent for a system to enhance digitized images. That claim was found to have a ‘technical effect’. Although the facts of this claim were superficially similar to those in Raytheon, it seems likely that the critical distinction between the two cases is that the input to and output from the device in VICOM corresponded to something tangible — the device took the input image and enhanced it. A telling passage from VICOM which was cited by the judge, Jeffs QC, notes that the claim in VICOM related to:

‘real world’ activities which start in the real world (with a picture) and end in the real world (with a picture).

This decision thus has implications for inventors of expert systems, whose output tends to be intangible information only.

It is a pity that UK cases have not done more to clarify exactly what it is that would render a computer program related invention patentable or unpatentable. Until they settle once and for all the role of ‘technical’ character in determining whether a claim relates to an invention, and what that ‘technical’ character involves, lawyers and patent agents can hope for little assistance from the decided cases.

125 For instance, the phrase is used in Diamond v Diehr, 450 US 584 (1981).
IV. RECENT DEVELOPMENTS: AUSTRALIA

The provisions of the Australia Copyright Act\textsuperscript{128} relating to computer programs are very similar to ours, so copyright decisions from this jurisdiction must have significant persuasive value here. Australia’s Patents Act 1990\textsuperscript{129}, however, is significantly different from our Patents Bill. Like American patent legislation, it lacks express statement that computer programs alone are unpatentable subject matter, but the Australian courts have developed rules to that effect, relying largely on US authorities.

a. Copyright Law

(i) Reform

The major development here is the impending report on the law relating to computer software protection by the Australian Copyright Law Review Committee. The Committee presented its draft report last year, and the final report is expected to be published before October 1994. If the Australian Copyright Act is amended in response, that will have the effect of making developments in Australia less significant here than they are presently. The draft report was not available at the time of writing, but there are a number of articles which describe and discuss its contents\textsuperscript{130}. Summarising briefly from these sources, the major features of the draft report are set out below.

(a) The Committee affirmed the choice of copyright as the most appropriate form of protection for computer programs for purely pragmatic reasons: so that Australia can keep in step with its major trading partners.

(b) On the protection of non-literary aspects of computer programs, the Committee concluded that structure, sequence and organisation of program code should be protected by copyright, but user interfaces should not. The latter, it was suggested, was behavioural so protection of a computer’s user interface would involve protection of the function of a program.

(c) Copyright in computer programs should extend to cover rental of copies to the public.

(d) Copying incidental to the normal use of a computer program, back-up copying and reproduction in the course of debugging

\textsuperscript{128} Act No 63 of 1968, as amended by the Copyright Amendment Act 1984 (Cth) (No 43 of 84).
\textsuperscript{129} Act No 83 of 1990.
where a properly functioning version of a program is not reasonably available should be exceptions to the copyright owner’s exclusive rights.

(e) Specific provision should be made to deal with circumvention of copy-protection and for the application of copyright law to computer-generated works.

(f) The Committee also recommended a limited decompilation right for programmers seeking to write interoperable programs or to correct errors in the operation of a program along similar lines to those set out in the EC Council Directive on the legal protection of computer programs.\(^\text{131}\)

(g) On parallel importation of computer programs, the Committee was divided; a bare majority felt that parallel imports should be allowed only from those countries which are the main sources of computer programs used in Australia. It was also recommended that in infringement proceedings, the burden of proving that imported programs were not pirate copies should fall on the importer.

(ii) **Autodesk Inc. v Dyason (No 2)**

The decision in *Autodesk v Dyason (No 1)*\(^\text{132}\) is well enough known that it is not necessary to go into the full details of that decision\(^\text{133}\). A development worth noting is a reconsideration by the High Court of Australia of its earlier decision, which is reported as *Autodesk v Dyason (No 2)*\(^\text{134}\). Although the result was the same — the finding of copyright infringement of the plaintiff’s AutoCAD program by the defendant’s device which circumvented an anti-piracy feature of the program was upheld — there are hints in the later decision that some of the judges have stepped back from somewhat extreme positions which they adopted in the earlier decision.

In *Autodesk (No 1)*, the Plaintiff’s program, ‘AutoCAD’, was designed to function only when a hardware device, called the ‘AutoCAD lock’, was attached to the computer. This was an anti-piracy feature of the plaintiff’s program; users could make as many copies of the program as they wished, but the program would not run unless an AutoCAD lock device was in

---

131 *Supra*, note 72. The major features of the UK decompilation right are described in the main text accompanying notes 79 to 81.
132 (1992) 22 IPR 163.
133 Reference may usefully be had to a number of articles commenting on this case such as K Manwaring, “Autodesk v Dyason” (1992) Sydney LR 518; A Bodger, “Software Protection in Australia: the Autodesk Case” (1992) 14 EIPR 211 and Prescott, “Was AutoCAD Wrongly Decided?” (1992) 14 EIPR 191.
place. This AutoCAD lock device could not be as easily duplicated as the program disks, and only licensed users would be able to obtain this hardware device legitimately, so it represented the copyright owner’s main protection against piracy.

The locking system involved a program within the AutoCAD package, called Widget C, which sent periodic signals to the AutoCAD lock. Hardware within the AutoCAD lock would process the signal and generate a response which would be transmitted back to the computer running the AutoCAD program. This response would be scrutinised by the program; it would be compared with a library of appropriate responses in what was called a ‘look-up table’. If the response received by Widget C was correct, the program would continue to function. But if there was no response (where for instance no AutoCAD lock was attached to the computer at the time) or the response was inappropriate, Widget C would cause the program to cease operating.

The infringement claim arose because the defendants designed and marketed a device, the ‘Auto Key lock’, which could perform the same function as the AutoCAD lock; it too could generate the correct responses to challenges issued by the AutoCAD program, thus enabling the AutoCAD program run without need for the AutoCAD lock. In other words, unlicensed users who could obtain a pirate copy of the AutoCAD program and the Auto Key lock could circumvent the locking system and use the program.

The Auto Key lock was developed by one of the defendants who had analysed the responses generated by the AutoCAD lock and, recognising that there was a repetitious pattern to the challenges and responses, worked out a table of appropriate responses. A copy of this table was stored in a memory device in the Auto Key lock. When the Auto Key lock was attached to a computer running the AutoCAD program, it would receive challenges issued by the program and refer to the table in its memory for the appropriate response. This response would then be transmitted to the program which would then continue to run. The defendants’ device thus had a very different nature from the AutoCAD lock; it generated the appropriate responses not by performing logical operations on the challenges but by sending out a response from its memory.

There was some controversy in the courts below whether the AutoCAD lock was itself a ‘computer program’ but the High Court in Autodesk (No 1) side-stepped this question and ruled instead that infringement was not of a computer program embodied in or represented by the AutoCAD lock, but of the look-up table in Widget C, which was duplicated by the table in the defendants’ Auto Key lock. The Court found that the look-up table had been ‘copied’ by the defendant and represented a ‘substantial’ part of the Widget C program, so the Auto Key lock infringed the plaintiff’s copyright in Widget C.
It was this finding — that the look-up table was a substantial part of the Widget C program — that led to academic controversy and which was challenged at the re-hearing in *Autodesk (No 2)*. There are two major criticisms that may be made of the original decision:

1. The look-up table could not be a ‘substantial’ part of the Widget C program because it consisted only of data; it may not even have been part of the copyright work. Even if it was part of the program, copyright could not subsist in that table itself because it was not ‘expression’, but merely a table of uncopyrightable facts.

2. Although not expressly stated, Dawson J, in his leading judgment with whom the other judges in *Autodesk (No 1)* agreed, seems to have determined that the look-up table was ‘substantial’ by considering its role in the functioning of the program; was it important or essential to the program’s operation? Such an approach is far too wide; it effectively means that any code whose absence would impair the operation of a computer program is a ‘substantial’ part of that program.

In *Autodesk (No 2)*, Mason CJ and Deane J, who had agreed with Dawson J in *Autodesk (No 1)*, conceded that the criticised parts of the reasoning ought to be reconsidered, but a majority held on other grounds that the matter should not be re-opened. This decision leaves us in the unsatisfactory position of having doubt cast upon the correctness of the major Australian copyright decision to date, but without a final resolution of the controversial issues raised. That resolution must await some later Australian High Court decision, or perhaps the Final Report of the Copyright Review Committee.

b. Patent Law

The basic consideration affecting patentability of computer programs is the common law principle drawn from US cases that a claim reciting only a mathematical algorithm is unpatentable. Two recent cases show that there is no single clear test in Australia for determining whether a computer program related claim is patentable or not. The first case, *IBM Corp v Commissioner of Patents* involved a claim for an improved method and apparatus for representing curved images in computer graphics displays. The invention essentially involved logical operations by a computer implementing known mathematical methods, and the ultimate result was the depiction of smoother and more accurate curves more quickly by the computer than with existing devices. The Commissioner refused registration, ruling that the claim was for nothing more than an algorithm, but the
Federal Court of Australia allowed IBM’s appeal, remitting the matter for reconsideration.

In his reasoning, Burchett J considered the major American cases in this area\(^\text{136}\) but ultimately decided that the test of whether the claim was patentable was to ask whether it resulted in a “commercially useful effect”. This test is based upon the Australian High Court decision in *National Research Development Corp v Commissioner of Patents*\(^\text{137}\).

This test had the potential to be the local paradigm in this area; it was recently adopted by the Australian Patent Office in its guidelines\(^\text{138}\) and would have represented a step away from dependence on US jurisprudence in this area of the law\(^\text{139}\). It is to be wondered, however, whether the ‘commercially useful effect’ test is suitable for general application to all computer program related inventions. The phrase ‘commercially useful’ has potentially a very wide meaning, probably going beyond the phrase ‘technical effect’. A system allowing a computer to process Chinese text could be described as being ‘commercially useful’ since it would make computers implementing that system more attractive to potential customers, but unless it represented a significant advance in computer hardware or computer technology, it would likely fall short of having a ‘technical effect’.

Perhaps recognising the danger inherent in this test, Cooper J in the other case of note, a later Federal Court of Australia decision of *CCom Pty Ltd v Jiejing Pty Ltd*\(^\text{140}\), declined to apply the test and held that a computer system for assembling Chinese language text was not patentable subject-matter. Reviewing the UK and US authorities as well as the *IBM* decision, his Honour stated his view of the law with the following propositions:

(a) Whether or not the subject matter of a claim constitutes a patentable invention is a question of fact.

(b) The mere programming of a standard computer with a computer program does not of itself constitute the computer as programmed a patentable apparatus.

(c) A computer program which constitutes an advance in the art of computer hardware or computer technology as such can constitute a patentable invention.

---

\(^{136}\) In particular, he considered the US Supreme Court decisions in *Gottschalk v Benson* 409 US 63 (1972), *Parker v Flook* 437 US 584 (1978), and *Diamond v Diehr*, supra, note 125.

\(^{137}\) (1959) 102 CLR 252.


\(^{139}\) For instance, another fairly recent case dealing with the patentability of a computer program related invention, *Re Application by Honeywell Bull Inc* (1991) 22 IPR 463, depends almost entirely upon a consideration of reported US decisions.

\(^{140}\) (1993) 27 IPR 577.
(d) A computer program which itself contains the steps in a process of new manufacture or forms part of a series of steps in a process of new manufacture can constitute a patentable invention.

(e) The language of the claim, whether it be a claim for a method contained in a computer program or a claim for an apparatus being a standard conventional computer operating in a conventional way to achieve a particular result by the use of a programmed ‘means’, will not be determinative of the patentability of the invention claimed and the court will look at the substance rather than the form of the claim to determine the true subject matter of the invention claimed and whether that subject matter is patentable.\(^\text{141}\)

This formulation does not leave any room for a general principle on the lines of the ‘commercially useful effect’ test suggested in IBM. Indeed, Cooper J explained the decision in IBM to have fallen under his proposition (c); the claim was patentable as ‘an advance in the art of computer hardware or computer technology’\(^\text{142}\). The claim in the instant case was little more than a mental process performed on a computer, and thus unpatentable. His Honour also fortified his conclusion by noting that the German Federal Patent Court considered a similar claim in \textit{Re the Computer Generation of Chinese Characters}\(^\text{143}\) and held it unpatentable since it lacked ‘technical character’. Although the judge professed not to have been unduly influenced by that German decision, it is clear that the reasoning employed in arriving at this conclusion bears the fingerprints of the European ‘technical character’ analysis rather than the US approach\(^\text{144}\). The difference in reasoning between these two cases means that a clear direction in Australian patent law in this area has yet to emerge.

V. CONCLUSION

The outcome of any survey in this area leads unavoidably to the conclusion that much confusion exists in the law between jurisdictions and within as to the extent and nature of the legal protection conferred upon persons who write or seek to exploit computer programs. The law of copyright in literary works is particularly confused because that regime is inherently unsuitable for dealing with the functional, dynamic nature of computer

\(^{141}\) \textit{Ibid} at 592.

\(^{142}\) \textit{Ibid} at 591 and 594.

\(^{143}\) [1993] FSR 315

\(^{144}\) No mention was made of an earlier decision of a Delegate of the Australian Commissioner of Patents on very similar subject matter in \textit{Thomas v Chappel} (1991) AIPC 90–798. Although the claim for extension of the term of the petty patent failed in that case on other grounds, the Delegate applied the test in a US case, \textit{Re Freeman} 573 F2d 1237 (1978) and said that the subject matter was not unpatentable.
programs. The confusion is exacerbated by the need for judges to properly understand the technology when they deal with such issues, and to set out in as clear terms as possible the analysis which must necessarily be performed by experts assisting the courts. Since our copyright law, like that of the UK, permits independent creation and some use of another’s work short of taking a ‘substantial part’ of the work, it will be left to the courts to attempt to explain when use of another’s work amounts to infringement. It is unlikely that a single clear test can ever be articulated to definitively solve the essentially policy-based questions in this area given the range of types of computer programs and their varying degrees of complexity, but courts must nevertheless work towards the most logical analysis consistent with the substantive and procedural rules of law of that jurisdiction. The answers to the essentially policy-based dilemmas may come through legislative intervention. The EC Directive on the Legal Protection of Computer Programs at least attempts to answer some of the questions but whether the position adopted is the best possible is still open to debate.

Patent law is similarly beset with confusion, though perhaps of a lesser degree. The question here is when the use of a computer program in an invention disqualifies that invention from patentability. The latest decisions from the jurisdictions under review are not completely incompatible; they seem to be coalescing around a notion that a computer program related invention will be patentable if it does more than use purely abstract imputs and produce a purely abstract result. The confusion primarily revolves around the terminology that must be used to define this notion, so the task of clarifying the law in this area is perhaps less daunting than for copyright.

CHRISTOPHER LEE GEN-MIN*

\[Supra, \text{ note } 72.\]

* LLB (NUS), LLM (Lond), Advocate & Solicitor (Singapore). I would like to thank Professor Gerald Dworkin, Associate Professor George Wei, Mr Terence Tan and Mr Daniel Seng for their generous assistance and comments in the preparation of the paper on which this article is based. All errors are mine.