



## 從比較法上觀察法官應如何在審判中審理 電腦程式著作間具有”實質相似”之證據資料

鄧迺騰\*

### 壹、前言：

原則上在一般著作權侵害的案件中，原告必須證明二件事，其中第一項是「提起訴訟之原告，其據以主張權利的著作，係一個著作權法上適格的、受保護的著作」。第二項則是「必須證明”被告抄襲”了該被著作權法保護之著作」。一般來說原告爲了證明前述第 2 項要件有 2 個方法，其中第一個方法是原告提出直接證據（direct evidence）證明被告抄襲。而第二個方法是原告提出證據來證明被告或有”接觸（access）原告之創作物”或”被告的作品與原告的作品二者間具有”實質的相似性”（substantial similarity）<sup>1</sup>。

### 貳、引發本文爭議點之所在：電腦程式設計之方法與流程

傳統著作權法針對文字著作的保護，係以文字著作爲基礎而漸次發展成現今的保護體系，爲了探討「電腦程式著作」在著作權法上，是否因爲其係具有不同於一般我們對文字著作的認識，而應該在著作權法上有不同的評價、並且在程序法上也應該有不同的證據調查方法？在回答這些問題之前，我們先必須就「電腦程式著作」的產生作一個瞭解。而

\* 現任浩瀚數位股份有限公司法務專員

<sup>1</sup> 請參照 羅明通教授 <著作權法論> 台英國際商務法律事務所，2002.8 第 4 版，第二冊第 301 頁。



## 論述

### 從比較法上觀察法官應如何在審判中審理 電腦程式著作間具有”實質相似”之證據資料

所謂撰寫一個電腦程式<sup>2</sup> 就是為了使電腦硬體設備在完成一定的工作前，我們必須先行撰寫讓電腦硬體據以執行所交付的工作而供其判讀的程式，而一般說來開發電腦程式的步驟如下：

第一步：

確定一個程式其主要功能。

第二步：

程式設計者依據該被特定的功能，把該功能加以解構 (de-composes) 成簡化的“各種次要問題”或者是”次功能” (simpler constituent problems subtasks)

第三步：

將”次要問題”加以排列組合，而形成一個”組織架構圖” (organizational map) 或”流程圖 (flow charts)”。

第四步：

為了使一個程式結構能夠更精確化，程式設計者會使該程式具有最快的運算速率。許多時候程式設計者將一個 macro 併同一個參數表的指示，來精確化一個指令，基於使一個結構能夠更精確，達成運算速率極大化、執行效率最佳化的要求，因此有時在這個階段必須考量到外在因素所造成的限制 (如記憶體的限制、該程式所欲搭配的電腦廠牌等等)。

第五步：“編碼”

也就是在考慮每一個模組間的相互關係，並且在概念上加以佈局 (conceptually lay out)後，接下來選擇一個使電腦能夠讀取該模組的電腦

---

<sup>2</sup> 請參照 COMPUTER ASSOCIATES INTERNATIONAL, INC v. ALTAI, INC. 982 F.2d 693; 1992 (LEXIS 33369) 判決 (以下簡稱 ALTAI 案) [http://www.lexis.com/research/retrieve/bVtz-zSkAl&\\_md5=80b28c644ebed5e924cd37d82bb8e69c](http://www.lexis.com/research/retrieve/bVtz-zSkAl&_md5=80b28c644ebed5e924cd37d82bb8e69c) (2003/9/29)，其中第一部分就電腦程式設計之技術背景說明”BACKGROUND-I. COMPUTER PROGRAM DESIGN”。



語言，也就是稱做"coding"。這個步驟又區分成 2 個階段

1. 把程式結構藍圖 (program's structural blue-print) 轉換成原始碼"source code"，這個步驟就像小說家爲了傳達他的概念，以文字語句來使一個輪廓更加具體<sup>3</sup>。
2. 一旦原始碼完成後，接下來就是將原始碼轉譯 "compile"成"目的碼 Object code"。所謂目的碼 Object code 是一連串由 0 與 1 組合而成的二進位語言，一種讓電腦得據以直接讀取指令之語言。

#### 第六步：除錯(de-bug)

完成編碼後，程式設計者會進行一連串的測試以及除錯( correct logical and syntactical errors)程序，也就是檢查其所寫出之程式中有無”邏輯”以及”語句上的錯誤”，直到 "de-bugging"這個步驟完成，才能被稱作是一個程式的完成。

### 參、傳統上基於著作權法立法目的，例外不應賦予著作權的判斷原則

著作權的保護，其前提就是「公共領域是從這些創造性的活動當中受有利益」，但是另一方面也因為基於其”立法目的”以及”公共政策與公益”上的考量，其並不保護一個”概念”或者是”想法的本身”，而是只單就「一個概念的實體上的表達方式」來提供法律上的保護。因此學者提出一些學說，來區分一個著作中「可受保護的部分」以及「性質上不應賦予排他權」的部分。茲將該等區別方法敘述如下：

<sup>3</sup> 請參照 ALTAI 案判決之”技術背景說明”，其中審理法官引用電腦學者 Kretschmer 文章的說法，其敘述「原始碼能夠以任何一種電腦語言來寫出來(COBAL, FORTRAN, BASIC, EDL)，其所差別者在於”該程式是為了要用在某一種特定的電腦上”。」



一、概念(思想)”與”表達”二分法 Idea /Expression Dichotomy<sup>4</sup>：著作權法的基本原則是「保護表達的方式」而不是保護「觀念本身」<sup>5</sup>，這個方法也可以用來決定著作權保護一項著作的界限。但是我們常常會遇到一個難題，就是法院於實際案例中很難去區分二者之界限。舉例來說，如果把”電腦程式”在「演算法」這個架構下來分析觀察，法院將很難就一個程式概括的做出結論，來決定在該程式當中哪一個程式所使用的演算法，才是著作權法上值得保護的表達方式。

二、概念(思想)”與”表達”合併原則<sup>6</sup>(The Merger Doctrine of Idea and Expression) 稱”概念”與”表達”合併原則就是指當某種觀念或思想只能被迫用一種方式來表達，或者是能夠表達的方式已經極其有限時，那這個表達的形式就不是受著作權法上保護的適格標的。因為這個時候該表達方式其實就是”表達那個觀念或思想的本身”，而觀念和思想並不是著作權保護的對象，所以我們說這個時候”觀念”和”思想”已經和其”表達方式”合併在一起。此時形式上所存在的各種不同的表達方式，都僅是傳達那個觀念和思想的本身，若對其中一個表達的方式提供其著作權法上排除他人使用的權利，就如同限制了其他人所創作出”其他的、可能的”表達方式的權利，而把原本屬於公共領域(public domain)中，由大眾所得分享和利用的”觀念”和”思想”，劃歸到私人所有的領域內。

<sup>4</sup> 請參照 John Swinson <哈佛科技法學評論 (5 Harv. J. Law & Tec 145) > “Copyright or Patent or Both: An algorithmic approach to computer software protection” [http://www.lexis.com/research/retrieve/frames?\\_m=1b40725bc4d8da3bfb7670cae7c4097&csvc=fr&cform=free&\\_fmtstr=XCITE&docnum=1&\\_startdoc=1&wchp=dGLbVtz-zSkAl&\\_md5=b42453ec7e93419a5272bea3e9c3666b](http://www.lexis.com/research/retrieve/frames?_m=1b40725bc4d8da3bfb7670cae7c4097&csvc=fr&cform=free&_fmtstr=XCITE&docnum=1&_startdoc=1&wchp=dGLbVtz-zSkAl&_md5=b42453ec7e93419a5272bea3e9c3666b) (2003/9/29)

其中第三部分關於著作權的保護範圍，其原文為「the scope of protection-Expression are protected, not ideas」亦即所保護的是表達的方式不是保護概念本身。

<sup>5</sup> 「概念與表達二分法」係 1879 *Baker v. Selden* 案, 101 U.S. 99, 25 L. Ed. 841，法院基於美國著作權法上之立法目的所發展出來的原則，其判決中文介紹可見 羅明通教授 <著作權法論> 第 253 頁以下。

<sup>6</sup> 請參照 John Swinson 前揭文第三部分，關於著作權法上二個沒有幫助作用的學說”B.2 unhelpful doctrines-2”Merger”部分的說明。



## 肆、依據電腦程式設計之特性，我們應如何界定電腦程式著作在著作權法上擁有的排他權範圍

### 一、概括賦予一個程式著作權的危險-以演算法為例

#### (一) 演算法 (Algorithms) 的定義：

演算法是一連串告訴電腦信息處理器，其應該如何完成一項被交付的工作的步驟。一個演算法能夠在不同的階層上，使用相異的表達方式，以不同的方式被表達出來。演算法的運算步驟必須以電腦訊息處理器能夠了解，並且據以執行的方式來撰寫，因此每一個電腦程式至少是一種演算法的表達方式之一。其可以被用來解決許多問題，而不是只用來解決數學問題本身。

#### (二) 就”演算法的性質”與其”相對應之著作權法上的評價”：

演算法從程式設計人員的角度來看，其可以”檢驗”並且”決定”這些解決方案中的異同，也因此法官適用著作權法來決定一件程式著作的權利範圍時，必須區別何者之電腦程式是基於不同的演算法，而以不同的語言在不同的抽象層次上來表達，而應該認為是對一個相同概念的不同的表達方法，而賦予不同的著作權法上的評價。

換言之何者之電腦程式是本於相同的演算法，而且是相同的表達方式，其係表達相同的概念？亦即其所產生的問題就是「演算法在選定特定處理器的時間點」會使得「演算法本身在著作權法上產生不同的評價結果」<sup>7</sup>。因此學說上會產生一個爭議，就是在哪一個時間點的演算法

<sup>7</sup> (甲說)一個以高階語言作為表達形式而寫出來的，能夠被一般的機器來執行的一連串指令，不算是演算法，頂多只能稱為其本質上是一個數學演算的公式 或者是一個觀念。一個演算法其抽象化的程度越高，則該處理器所需要指定使用的裝置的”被特定性就越低”。因為一個演算法必須和一個特定的裝置結合在一起，之後才能共同被使用在一個問題的處理程序上。因此一個以高階語言作為表達形式而寫出來而能夠被一般任何的機器來執行的一連串指令，就只能被稱為是”一個觀念 idea。”所以在能夠具體指定這個設備之前，都不能架構出一個完整的演算法。而這個時候不能賦予著作權法的保護。  
(乙說)一個處理器至少要被”具體定義和一個特定的演算法相容”或者”被期待是要



應當被評價為「著作權法的概念、思想本身」而不應該賦予著作權的保護；而哪一個階段的演算法已經算是「具有個人創作性的表達方式」而應當到受著作權的保護。因此法官在決定提供著作權法上的保護時，必須先對演算法加以了解，並且依照演算法之性質選擇應該賦予其在著作權法上有如何的評價。

原則上對於相同演算法可以依照對一個概念從「具體」到「抽象」區分為「低階機械語言層次上的精確化的表達」或者「是針對一個問題的高階語言的解決方案的描述」，雖然二者只有在技術層面上不同，也就是只有「表達一個概念的精確度」的不同而已，但是二者在著作權法的層面上，依照前述的「合併原則」就會有不同的評價。亦即我們必須區分二種情形來說明。(a) 在「抽象層面的針對一個問題的高階語言的解決方案的描述」，就是著作權法上的”概念和思想本身”，其不能為特定人所獨佔，因此不能賦予其排他性的權利；(b) 而對於相同演算法在「”低階機械語言層次上精確化的表達”」，就應該認為是表達方式的一種，原則上這個部分就可以受到著作權法的保護。我們接下來就這二個部分來詳細說明：

### 1、就「程式設計初期階段」而言：

依照演算法在”數學方面”以及”程式設計上的性質”而言，演算法是獨立的存在。如果我們以相同的演算法能夠被許多不同的處理器來執行的這種角度來看，原則上每一種演算法都可以適用在不同的電腦處理器上；相同的演算法原則上都能夠用二種以上不同的語言被表達出來，此

---

和某個特定的演算法相配合來運作”。而只有在”特定所要搭配的處理器”以及”為了該特定處理器的特性而選定最適合撰寫的程式語言”之後，這個演算法才會被具體特定的。這個時候的演算法就已經算是一個概念的表達方式，而應當受到著作權法的保護。

以上二種學說的爭議點是閱讀 John Swinson”前揭文”後，依照其對於演算法的特性的描述，參酌著作權法上的「概念與表達二分法」後，晚見認為在侵害他人著作的訴訟案件中所可能產生之「如何認定可受著作權保護的演算法」的認定標準的爭議，並其分別在訴訟程序上可作不同的實體抗辯與主張。



時雖然表達的方式不同，但是其「基本觀念 (idea)、概念 (concept)、構想 (plan)」等如何解決這個特定問題的「一般功能性的解決方案的描述」以及「這個程式最終的運算結果」將會是相同的。

因此在程式設計的初期階段，處理器不需要被具體指定，一個電腦科學家在設計程式初期「使用一般符號」，或者是「流程圖」針對所要完成的工作，表達其預定要使用的演算法。在這個階段，演算法是用「高階的符號形式」表達出來，此時在程式設計者心理中不一定先行設定所欲搭配使用的處理器，只是在使用演算法的過程當中，都至少已經有一個被期待要與該演算法和其相容的處理器，使之相配合來完成一定的任務。因此在這個階段中「演算法」因為屬於在「抽象層面的針對一個問題的高階語言的解決方案的描述」，所以在著作權法上的評價上，「演算法」就是著作權法上的「概念和思想本身」，其不能為特定人所獨佔，因此不能賦予其排他性的權利。這是適用「概念與表達二分法的結果」。

## 2、就「程式設計後期階段」而言：

正因為「表達的層面」取決於被選定的電腦語言複雜性的程度，一個電腦程式只是一個演算法在具體化的過程當中眾多表達方式的一種，亦即如同前述的在程式設計流程當中談到的，一個演算法可以用不同的方式來加以表現，或者以不同的抽象化的程度來加以了解。因此當我們面對一個待處理的問題，要求程式設計者提供一個更細節性的解決方案時，這個解決方案就逐漸從抽象的層面不斷的具體化，而由程式設計人員使用相異的表達方式來撰寫。而程式設計師為了不斷精確化一個演算法，在這個階段中，其將會以一個不具有顯著變化的二進位電腦語言，用機械碼 (machine code) 針對「邏輯」以及「語法」上來作修改，這個部份具有「思想的創造性」，因此雖然對於使用相同的演算法，但是在「低階機械語言層次上精確化的表達」的階段上，就應該認為是屬於表達方式的一種，而原則上這個部分就可以受到著作權法的保護。

### (三) 概括賦予一個程式著作權的危險

#### 1、運用「概念與表達二分法」賦予一個程式著作權的危險：



當能夠明確區分「概念」以及「表達方式」時，此時以「概念與表達二分法」作為前提，在適用該原則來判斷電腦程式著作中何者為概念本身；何者為表達方式的標準就是「一個程式所使用的演算法，就是這個程式的概念、構想，其作為該程式的基礎，這個部份不能賦予著作權的保護；而程式本身就是這個概念的表達方式，這個部分才是著作權法要保護的對象」這個說法太過於簡化，會造成不合理的結果。

既然一個特定的結果，可以基於一個相同的演算法在不同的具體化的階段上來做表達，因此「以功能取向作為賦予著作權的方式」就會發生一些不合理的現象。詳言之，一個電腦程式和另外一個程式彼此都有相同的功能，而一個程式的功能就是這個程式的目的，對於”相同的功能”在「抽象層面」上可能是以不同的演算法來達成，也可以用相同的演算法來表達。對於用不同的演算法所具體化後的程式表達當然不會相同，此時二個演算法之間沒有所謂「實質相似」的問題，所以在這個部分當然不能就做出「二個程式所解決者是一個相同的問題，認為具有相同的功能，就概括的先行賦予其中一個程式在特定的功能上有一個著作權，而認為其他具有相同功能的程式都是落入這個受著作權法保護的範圍之內。」這樣的結論，因為其可能”擴張”了著作權人受到保護的範圍。

### 2、運用「合併原則」賦予一個程式著作權的危險<sup>8</sup>：

當不能夠明確區分「概念」以及「表達方式」時，把「演算法」認為是屬於概念本身，此外基於「效率的追求」、「外在硬體的搭配」等等的限制使得「表達該演算法的方式因為是極其有限的」而有「合併原則」的適用，一概否認該程式有著作權的這種的判斷也是有問題的。與前一小段敘述的角度相反，此時在「具體的層面」上，即便使用相同的演算法，由於程式間的不同之處就是在於其”以達成所要解決的工作為目的而如何的來使用演算法，而每個程式設計師對於一個概念的具體化描

<sup>8</sup> John Swinson 在前揭文當中認為，法院創設出的”合併原則”誤解了”概念/表達二分法”的意義，合併原則違背了著作權的體系。





述，便是其創意所在，如果對於一個相同的概念能有不同的表達方式出現，法院這時候如果仍然概括的否定任何一個程式在特定的功能上都不允許有著作權，而認為其他具有相同功能的程式都是「相同概念的相同表達方式」而係屬「概念與表達合併的結果」，這樣的結論也同樣會造成不合理的結果。

今天我們假設，有第三人是基於獨立創作而產生了和既有表達方式不同，而屬於另外一個對相同概念和思想的表達方式，這就說明了“合併原則”這個學說所能解決的問題是有限的，因為這個第二個創作出不同表達方式的人，正因為其表達方式不同，因此先前所認為的「概念與其表達方式合併」的前提就不成立，所以第二個人所創作出的表達方式就不算是侵害他人的著作權，而屬於另一個被著作權法保護的客體。

法院通常把“演算法”就當成是“觀念和思想的本身”，而忽略了我們之前對演算法的定義的瞭解，因為即使是“演算法本身”都可以在不同階層中，來作的不同抽象化程度的表達。沒有一個演算法可以被特定使用在一個程式當中，不同的程式設計師可以在任何一個抽象的程度上來陳述一個演算法，而每一個演算法的運算結果都可以完成該程式所要達成的目的。

一個程式會因著當初這個程式所被選擇來表達的那個概念的抽象程度而有不同的撰寫方式，我們不能概括的認定“這個程式就是表達某個特定概念的唯一的方式”，因為當一個設計人員從另外一個層次來看這個概念時，它就可以用另外一種方式來表達這個概念。

正因為所有的程式都將有一種以上的表達方式，為了不侵犯他人程式著作起見，設計人員會先以一個簡單的演算法的表達方式，來繞過原有的程式本身的表達方式，因此可以得知某個特定的概念的確有多種不同的表達方式存在著。

因此在適用「合併原則」的時候，我們應該先決定如何認定「表達方式的有限性或唯一性」，如果事實上當事人能提出證明，的確對於相同概念有超過一種以上的表達方法時，法院就不可概括的先就「合併原



則」加以適用，而認為「主張系爭程式有著作權的當事人其程式碼不受著作權的保護」，也就是說法院這個時候必須要去分辨哪些方式是「相同概念運用相同的表達方式」，而哪些程式算是「該相同概念的第二種表達方式」。在決定二種表達形式是否屬於實質相似的時候，就是要解決這個問題，所以合併原則只是陳述了一個結論，不是一個測試的方法，其可能”縮小”了著作權人受到保護的範圍。

### 二、如何適當的界定電腦程式著作的排他權的範圍-從比較法的角度看美國 Computer Associate. V. Altai 案的三階段分析方法

#### (一)Computer Associate. V. Altai 案的三階段分析方法

我們可從比較法的角度來觀察美國法院就 Computer Associate. V. Altai 案的審理方式，其如何於「侵害他人電腦程式著作案件」的審理過程當中，正確的基於著作權的立法目的，先結合各種學說來劃分程式當中「受著作權法保護」的部分，以及「不受著作權保護的部分」，進而判斷二個程式間「是否具有實質相似」的判斷方法<sup>9</sup>，作為我國法院日後審理相同案件之參考，學說稱該案採用的是「三階段分析」的方法，茲將各分析步驟簡述如下：

第一步驟「抽離」：訴訟當事人與法院指派之第三公正鑑定人將電腦程式解構，並加以抽象化。亦即把一個電腦程式的原始碼（source code）、目的碼（object code）、參數表（parameter list）、模組（module）、常數（routine）以及”一個程式最終功能目的的一般性描述（Ultimate functional purpose）”把一個電腦程式在不同階層上加以區分為不同的表達形式，並將其做成表列。

<sup>9</sup> 請參照 ALTAI 案判決，上訴法院認為：一個程式一定有「受到保護的部分」以及「不受到保護的部分」，先區分被告的程式中未含有原告的程式中受到保護的部分，接著才需要去考慮”實質相似”的問題。其原文為「... the defendant's program structure might contain protectable expression and/or other elements that are not found in the plaintiff's program. Since it is extraneous to the allegedly copied work, this material would have no bearing on any potential substantial similarity between the two programs.」。



其思考方式是在這個不斷解構抽離的過程當中，會有一個程式任務的概念化模式出現，當該任務模式出現後，便判斷這個模式有沒有觀念上的普遍性（conceptual normality/ general notion），如果有普遍性存在，則將它歸屬於”公共領域（public domain）”的部分而被視為一個無法受著作權保護的”概念本身”。第二步驟「過濾」：

1. 由於著作權所保護的是”表達的方式”，所以一方面”一個概念本身”無法賦予其著作權而排除他人的使用所以必須把它加以濾除，也就是將原告所主張的受到侵害的電腦程式當中，對於解構後具有普遍性的抽象概念予以濾除，此時所運用的原則就如同前述之「概念與表達二分法」一般。
2. 在這個部分同時也運用了前段所述的「合併原則」，來過濾掉「概念」與「表達方式」基於外在的限制，使得表達方式具有「有限性」<sup>10</sup>，而不應該獲得著作權保護的部分。

第三步驟「比較」：依據前二階段所濾除者都是不受著作權法保護的部分，則濾除後所剩下的部分就是屬於「受到著作權保護的表達方

<sup>10</sup> 請參照 ALTAI 案判決，此時所據以為「基於電腦產業的特性所無法賦予其排他性的著作權」而應當排除著作權法的保護而認為「表達」與「概念」合併，來限縮「主張該電腦程式之人其擁有著作權的範圍」之理由如下：

(a)基於「電腦產業的特性」：雖然

在進行電腦程式的開發時，每一個程式都有其可資適用的演算法。

一個程式的功能通常都有一個以上的演算法在不同的抽象化程度上來加以表達，但是各種方式卻有不同的「效率」與「運算速度」，為了達成「效率最佳化的目的」，就會有合併原則的適用。

(b)對於「程式設計的外在限制（external constrain）」：

除了”演算法上的效率追求”之外，還有電腦產業中對於「硬體規格的標準化 mechanical specification」所對於軟體設計的限制，例如只能以特定指令驅動特定廠牌的電腦內之特定功能。

(c)基於「相容性」的限制：應用程式也受到該程式所預計要與之相配合的作業系統程式（operating system）與其他介面（interface）程式相容性的牽制。

(d)「既定商業模式」之限制：若是對特定產業所撰寫的程式，由於該產業對特定程式的實際需求有其商業上的習慣，原則上針對同一商業行為所設計出來的程式，將不可避免的有極多的相似性。

(e)「撰寫軟體的習慣」所造成的限制：軟體工程師間會存在有普遍被接受的程式撰寫慣例，亦即對於解決一般問題所共同會使用的解決手段，這個部分就類似於傳統著作權法上判斷「實質相似」的”必要場景”原則一樣，對於普通主題 這個部分的相似，因其欠缺個人的創作性與個人技巧，所以不應被評價是屬於著作。



式」，所以這個時候才將「原告宣稱有著作權保護的電腦程式」再與「被告被起訴抄襲原告著作的程式」加以相比較，視其是否具有「實質的相似性 substantial similarity」，此時如果有實質相似之處，則法院就必須衡量被告的程式當中其與原告程式相似之處，是否具有實質之重要性。亦即最後才考慮是否合乎「合理使用」的原則。

(二)以 Computer Associate. V. Altai 案觀察法官對於證據的認定：

1、就證據分析 ( Evidentiary Analysis ) 而言：

本案審理的地方法院如同前述之分析方法，是先做抽離分析，從一個”逐漸一般化的方式”亦即從目的碼、原始碼、參數表、目標功能以及一般整體架構在二個程式之間抽離、過濾後，從所餘的部分中，在每一階層來區分之間到底「有無實質相似的成分」<sup>11</sup>。上訴法院原則上同意地方法院該等證據分析之結果，亦即：

- (1) 被告程式 OSCAR 3.5 在抽象階層的結構成分 ( services required, general outline ) 上，在排除了原告不受保護的表達方式的部分後，原告在這個部分沒有權利可以作為請求基礎<sup>12</sup>。
- (2) 在”目的碼”以及”原始碼”上二者全然不相同<sup>13</sup>。
- (3) 對於參數表 parameter lists 與 macros 的部分：
  - a、二者一部分類似，但是這部分相對於程式的整體來說，其

<sup>11</sup> 請參照 ALTAI 案判決，其原文敘述為「The district court took the first step in the analysis set forth in this opinion when it separated the program by levels of abstraction. The district court stated: As applied to computer software programs, this abstractions test would progress in order of "increasing generality" from object code, to source code, to parameter lists, to services required, to general outline. In discussing the particular similarities.」

<sup>12</sup> 請參照 ALTAI 案判決，其原文敘述為「...OSCAR 3.5 structural components, we agree with Judge Pratt's systematic exclusion of non-protectable expression.」

<sup>13</sup> 請參照 ALTAI 案判決，其原文敘述為「With respect to code, the district court observed that after the rewrite of OSCAR 3.4 to OSCAR 3.5, "there remained virtually no lines of code that were identical to ADAPTER."」



性質上非該程式的重點，而份量上不構成程式主要的結構，所以都不能被證明為侵權。

b、其他部分係屬於公共領域（public domain）或者是屬於該程式其功能上的本質需求。

（4）流程圖的部分：該流程圖只是一個概念的表現，是一個功能主題，係針對該程式所要解決的問題的描述，就”必要場景原則”（the *scenes a faire* doctrine）<sup>14</sup>的判斷後，該部分也是不受保護的。

## 2、在 ALTAI 案第二上訴巡迴法院所提出之問題：

受理本案之上訴法院提出一個問題<sup>15</sup>，就是在審理過程當中，基於訴訟經濟的要求，應該在審判程序當中被過濾的程式，是被告的程式？還是原告的程式？也就是說「法院審理的重點」以及「審理的順序」，應該是先過濾原告的程式，以分析並確定原告的著作權權利範圍？還是應該先對被告程式的結構做抽離分析，直接看看是否有無和原告程式具有實質相似的部分？本案判決除了點出這個問題之外，對於應該如何解決這個問題並未做具體的處理。

晚見以為，如果就我國刑事訴訟程序的起訴階段而言，主張權利受到侵害之直接被害人，就被害事實之有無必須先做主張。亦即告訴人或自訴人其權利之有無及其範圍必須先行在客觀上來作形式上之認定，故必須先以原告主張之程式來作分析，以辨明其是否真有如其所主張之被害事實。而在具體訴訟中對「證據提出」、「證據的審理」和「訴訟程序應如何進行」的部分，將於下段中以我國法制為例，來提出一個可資適

<sup>14</sup> 所謂「必要場景原則」就是任何要解決這個相同問題的人，都會不可避免的從這個方式著手進行，所以不能賦予這個具有普遍性的主題一個排他權利使特定人予以獨占。

<sup>15</sup> 請參照 ALTAI 案判決，其原文敘述為「...Thus, its filtration would be wasteful and unnecessarily time consuming. Furthermore, by focusing the analysis on the infringing rather than on the infringing material, a court may mistakenly place too little emphasis on a quantitatively small misappropriation which is, in reality, a qualitatively vital aspect of the plaintiff's protectable expression.」



用的審理方式。

### 伍、從我國新修正之刑事訴訟法，來檢討法院應如何判斷“ 電腦程式著作間是否具有「實質相似」”的證據資料

#### 一、就「判斷二個電腦程式間是否有實質相似」這個事項應當交 由專家鑑定

在“公訴程序”和“自訴程序”中的言詞辯論期日，審判長於檢察官或自訴人陳述起訴要旨、就被訴事實訊問被告之後，就必須進行「證據調查」的階段，在這個時候檢察官或自訴人以及被告、辯護人等，必須針對調查證據的結果就「事實」以及「法律」辯論之。在這個階段中，審判長應當如何針對「電腦程式著作」的特性，來對被告以及檢察官或自訴人所提出之證據資料加以審理？

簡單的說，在審理被告是否抄襲原告之電腦程式著作，就原告部分而言，實行公訴的檢察官與自訴程序的自訴人必須針對著作權法第九十一條第一項「重製行為」或者同法第九十二條第一項的「編輯行為」或「改作行為」，先行提出證據證明被告有如上之行為而侵害原告之著作財產權。對於被告而言，就必須針對原告所提出之證據提出反駁，並另行請求調查有利於己的證據<sup>16</sup>，此時二造就犯罪行為之有無的爭執點，就會落在「二個程式著作之間有無實質相似」的判斷上，此時該等證據資料在證據的評價上，對於未受電腦程式設計訓練之法官而言便是一個問題。

（一）針對「判斷電腦程式間有無實質相似」這個事項「有無鑑定必要性」的判斷而言：

就「判斷電腦程式間有無實質相似之交付鑑定的必要性」判斷而

<sup>16</sup> 刑事訴訟法第一百六十一條之一規範被告之舉證責任，其內容是「被告得就被訴事實指出有利之證明方法。」



言，其涉及該等證據資料所欲澄清的待證事實是否屬於「法院裁量<sup>17</sup>」的部分，其中的區別實益就在於：是否該等證據資料是屬於法官所得自行做出證據判斷的事項，而法院可以獨立的依照「自由心證」原則，在不違背「經驗法則」以及「論理法則」的前提下，賦予其證據價值，此時沒有把系爭電腦程式交付鑑定，並不構成刑事訴訟法第三百七十九條第十款<sup>18</sup>的「判決違背法令」事由；還是說該等「據以認定被告重製、改作、編輯原告電腦程式著作」的證據，基於該證據資料的「高度專業性」，因此法官或一方面依其職權直接交由公正第三人加以鑑定，或經由當事人依照刑事訴訟法第一百六十三條來聲請調查證據後，再交付鑑定，此時法官在認定犯罪事實的有無之前，就必須讓當事人先對鑑定報告作事實與法律的辯論後，才做出法律上的評價。亦即這個時候沒有交付鑑定就會違背刑事訴訟法第一百五十四條之「證據裁判原則」以及沒有依照「經驗法則」以及「論理法則」，而濫用同法第一百五十五條第一項之自由心證主義，而構成刑事訴訟法第三百七十九條第十款的「判決違背法令」。因此對於這個「有無鑑定的必要性」的判斷方法，學者提出 2 個判斷的標準<sup>19</sup>：

1、該待證事實是否屬於「法院澄清義務」的範圍：此時所要斟酌的

<sup>17</sup> 在 ALTAI 案判決中上訴法院法官認為這是屬於法院的裁量權，請參照 ALTAI 案判決“evidentiary analysis”的部分，其中談到“就 2 個電腦程式是否屬於高度複雜性的事項而屬於一般人士可以做出判斷；或者是必須交由專家證人來決定，這是屬於法院的裁量權”。其原文敘述為“(applying the “ordinary reasonable person” standard in substantial similarity test for computer programs) Rather, we leave it to the discretion of the district court to decide to what extent, if any, expert opinion, regarding the highly technical nature of computer programs, is warranted in a given case. ”

<sup>18</sup> 刑事訴訟法第三百七十九條判決當然違背法令事由第十款「應於審判期日調查之證據未予調查」。

<sup>19</sup> 林鈺雄教授 <刑事訴訟法>元照出版公司 2000.12 初版 上冊 第 399 頁。



是該證據必須具有關連性、調查必要性和調查可能<sup>20</sup>。

- 2、該證據「必須經過特別專門知識之人的判斷」方能協助法院澄清待證事實。

就這二個標準來看本文所討論的主題，也就是對於「判斷二個電腦程式間是否有實質相似」這個事項的性質而言，因為在該等事項的技術性質，是屬於必須經過特別專門知識之人，也就是從事電腦程式設計、熟悉該產業間的人士，方得做出「相似與否」判斷，因此晚見認為該等事項並非屬於法院的澄清義務的範圍之內而「有交付鑑定的必要性」，此觀我國刑事訴訟法在修正第一百六十一條以及第一百六十三條改採「當事人進行主義」而法院只負擔「補充性」的證據調查義務之後更加顯然。

### (二) 鑑定人如何對「判斷二個電腦程式間是否有實質相似」進行鑑定：

原則上鑑定人對「判斷二個電腦程式間是否有實質相似」只做客觀判斷，不做法律評價，就「鑑定人地位」與「鑑定行為的性質」來說，「鑑定人」係本於其專門知識，輔助法院判斷特定證據之人，故其行為之性質是：

- 1、鑑定人係依照專門知識提供「一般經驗法則」：

就本文所討論的主題來說，就是提供審理的法官，對於一般電腦軟體程式師設計電腦程式時，其應有之步驟與其相關知識，就其在該產業上以一般的角度，對法院提出其鑑定之方式。

- 2、鑑定人係針對某個唯有依照「特別專門之知識」始能察覺、判

<sup>20</sup> 這三個要件可見於刑事訴訟法第一百六十三條之二關於法院認為不必要調查的證據的判斷方式請參照條文全文「當事人、代理人、辯護人或輔佐人聲請調查之證據，法院認為不必要者，得以裁定駁回之。下列情形，應認為不必要：

一、不能調查者。(調查可能性)、二、與待證事實無重要關係者。(證據關連性) 三、待證事實已臻明瞭無再調查之必要者。四、同一證據再行聲請者。」





斷之事實來作鑑定。而依照本文的爭議點就是鑑定人其係針對原告與被告之電腦程式，對於二個程式之間就「是否具有實質相似」這個部分單純客觀的表達其專家意見。

3、鑑定人是依照”科學論證規則”而做出一個唯有依照”專門知識”使能判定的結論，但是不能作出法律上之評價，亦即其不能作出法律上之結論。該鑑定結果仍必須經過法院的言詞辯論程序來檢驗。而在這個針對原告程式與被告程式做出鑑定的階段，訴訟程序的當事人一方面可參照本文所介紹的 Altai 案中二審法官所建議的「三階段測試法」，先行就自己的電腦程式著作，依照程式開發流程提出各種不同階層的該特定程式著作的不同表達方式的表列，並且提出自己延請的鑑定人所做出的鑑定報告。而審理的法官也可指定第三鑑定人，代替法院就各當事人提出之程式表列，針對原始碼、目的碼、參數表、模組、常數以及”一個程式最終功能目的的一般性描述”，等在每一個階層中做出客觀的「是否在每一個階層中有無實質相似」的鑑定報告。以便使法官進行下一個階段的程序，也就是將「調查證據的結果」，使當事人就事實與法律辯論，並使當事人對鑑定人予以詰問，而得出「全辯論意旨」。

## 二、法院與當事人就鑑定結果之處理方式

對於鑑定報告之處理，原則上法官不能只憑鑑定報告就直接作出法律上之評價，亦即鑑定之結果仍然必須經過「言詞審理」，法官方能就其是否該當實體上之構成要件，來做出法律上的評價，其屬法官之責任，因為法官要適用法律來確定國家對該違反著作權法的行為人其「刑罰權的範圍」，而負有對「構成要件的有無」有一個「澄清義務」，對於這個部分法官不能全部交由鑑定人為之。換句話說對於事實認定，法院並非全盤接受鑑定的結果，基於刑事訴訟程序的基本原則諸如「直接審理主義」、「言詞辯論主義」、「自由心證主義」和「證據裁判主義」等等，法院必須使其經過鑑定人就鑑定事項與鑑定意見讓當事人進行交互詰問、進行攻防後，得出對該書面鑑定報告的心證，並對該經過「證據調查程序」檢驗後的鑑定意見之取舍作出法律上之審酌，並將之記明判決



理由，如此才能使上級審針對事實或者是法律有事後審查之依據。

在依照新修正的刑事訴訟法採取「改良式的當事人進行主義<sup>21</sup>」後，對於訴訟程序的實施，一方面當事人可自行提出訴訟外第三人所做之鑑定報告，但是當事人也可以依照刑事訴訟法第一百六十三條以下之規定來聲請調查證據<sup>22</sup>，法院依據當事人之申請，再選定公正之第三人作為法院的鑑定人<sup>23</sup>，做出該二個程式在客觀上有無「實質相似」的部分<sup>24</sup>，

<sup>21</sup> 請參照刑事訴訟法第一百六十一條之二（當事人進行主義）  
「當事人、代理人、辯護人或輔佐人應就調查證據之範圍、次序及方法提出意見。法院應依前項所提意見而為裁定；必要時，得因當事人、代理人、辯護人或輔佐人之聲請變更之。」，而在 ALTAI 案判決中上訴法院法官就這個部分認為有無進行鑑定的必要屬於當事人主張權利的部分因此就該等事項，法院與雙方當事人同意交由公正第三人來鑑定，其原文敘述是「with the consent of both Altai and CA, Judge Pratt appointed and relied upon Dr. Randall Davis of the Massachusetts Institute of Technology as the court's own expert witness on the issue of substantial similarity.....」

<sup>22</sup> 刑事訴訟法第一百六十三條（職權調查證據）  
當事人、代理人、辯護人或輔佐人得聲請調查證據，並得於調查證據時，詢問證人、鑑定人或被告。審判長除認為有不當者外，不得禁止之。  
法院為發見真實，得依職權調查證據。但於公平正義之維護或對被告之利益有重大關係事項，法院應依職權調查之。  
法院為前項調查證據前，應予當事人、代理人、辯護人或輔佐人陳述意見之機會。

#### 刑事訴訟法第一百六十三條之一（調查證據之程式）

當事人、代理人、辯護人或輔佐人聲請調查證據，應以書狀分別具體記載下列事項：  
一、聲請調查之證據及其與待證事實之關係。  
二、聲請傳喚之證人、鑑定人、通譯之姓名、性別、住居所及預期詰問所需之時間。  
三、聲請調查之證據文書或其他文書之目錄。若僅聲請調查證據文書或其他文書之一部分者，應將該部分明確標示。

<sup>23</sup> 刑事訴訟法第一百九十八條（鑑定人之選任）  
「鑑定人由審判長、受命法官或檢察官就下列之人選任一人或數人充之：  
一、就鑑定事項有特別知識經驗者。二、經政府機關委任有鑑定職務者。」

<sup>24</sup> 請參照 ALTAI 案判決，使用專家鑑定意見做為決定電腦程式是否實質相似的方法  
「Use of Expert Evidence in Determining Substantial Similarity Between Computer Programs」；專家證人做出對於電腦程式實質相似與否的不同面向的分析並且針對原告與被告各自提出之專家證人的意見做出評論「Dr. Davis submitted a comprehensive written report that analyzed the various aspects of the computer programs at issue and evaluated the parties' expert evidence.」



接著使做成該「鑑定報告」之鑑定人於「證據調查程序」中出庭<sup>25</sup>，讓其就所鑑定的事實，與其做出之鑑定報告，由當事人來進行「交互詰問<sup>26</sup>」，以期追求「實體的真實發現」。最後再由法官針對「調查證據的結果」與「全辯論意旨」，對這個在著作權法上屬於判斷「重製、改作、編輯」等犯罪構成要件，或者是「是否構成實質相似」的不成文犯罪構成要件上，來做出”何者屬於著作權法保護的範圍”，”何者不屬於著作權法保護的範圍的法律上的評價<sup>27</sup>。”

<sup>25</sup> 刑事訴訟法第一百六十八條（證人、鑑定人之在庭義務）證人、鑑定人雖經陳述完畢，非得審判長之許可，不得退庭。

<sup>26</sup> 請參照刑事訴訟法第一百六十六條（對證人、鑑定人之詰問）  
「當事人、代理人、辯護人及輔佐人聲請傳喚之證人、鑑定人，於審判長為人別訊問後，由當事人、代理人或辯護人直接詰問之。被告如無辯護人，而不欲行詰問時，審判長仍應予詢問證人、鑑定人之適當機會」

刑事訴訟法第二百零六條（鑑定報告）

「鑑定之經過及其結果，應命鑑定人以言詞或書面報告。鑑定人有數人時，得使其共同報告之。但意見不同者，應使其各別報告。以書面報告者，於必要時得使其以言詞說明。」

刑事訴訟法第二百零八條（機關鑑定）

「法院或檢察官得囑託醫院、學校或其他相當之機關、團體為鑑定，或審查他人之鑑定，並準用第二百零三條至第二百零六條之一之規定；其須以言詞報告或說明時，得命實施鑑定或審查之人為之。第一百六十三條第一項、第一百六十六條至第一百六十七條之七、第二百零二條之規定，於前項由實施鑑定或審查之人為言詞報告或說明之情形準用之。」

在 ALTAI 案判決中，審判程序進行到證據調查階段時，法院指派之專家證人也必須接受交互詰問，其原文為「...At trial, Dr. Davis was extensively cross-examined by both CA and Altai.」

<sup>27</sup> 第一步：先決定二個程式之間相似的部分的量有多少，第二步：再決定該相同的部分是否違法。這二個步驟都是屬於事實發現的部分，對於第二步驟中的判斷是否實質相似又分為二個部分（1）先由一般非電腦專業 lay observers 之人士來判斷（2）接下來才由專家意見作鑑定一旦由前者就可以做出判斷則此時就不需要再做專家意見的鑑定，而使用專家鑑定意見的門檻，就是其可否協助法院發現真實。原則上對一個具有高度複雜性與技術性的事項，通常必須仰賴專家鑑定意見，作為認定事實的基礎。「As a threshold matter, expert testimony may be used to assist the fact finder in ascertaining whether the defendant had copied any part of the plaintiff's work. once some amount of copying has been established, it remains solely for the trier of fact to determine whether the copying was "illicit.... Since the test for illicit copying is based upon the response of ordinary lay observers, expert testimony is thus "irrelevant" and not permitted. Thus, in deciding the limits to which expert opinion may be employed in ascertaining the substantial similarity of computer programs, we cannot disregard the highly complicated and technical subject matter at the heart of these claims.」



## 論述

從比較法上觀察法官應如何在審判中審理  
電腦程式著作間具有”實質相似”之證據資料

三、我國法院就判斷「電腦程式著作間是否有重製、改作事實之有無」的判斷方式稍顯簡略---以台灣台北地方法院八十九年度易字第三四二號判決、最高法院九十年度台上字第五三一號判決中「判斷有無重製、改作電腦程式著作的證據調查」部分為例

(一) 就台灣台北地方法院八十九年度易字第三四二號判決<sup>28</sup>而言：

1、案件事實部分：

原告主張的部分：

其”車輛檢驗線軟體之電腦程式著作”(含DOS、UNIX、WINDOW三項作業系統)被告在未經原告授權時予以重製、改作而侵害原告之著作權係屬違法而提出告訴。

A. 該案扣押後所得之證據資料：

於臺北市監理處扣得的部分是「電腦主機硬碟」。

於山隆修車廠扣得的部分是「硬碟內UNIX作業系統程式列印資料一份」。

於東陽修車廠扣得的部分是「硬碟內UNIX作業系統程式列印資料九份，電腦程式磁片五片。」

B 被告抗辯的部分：

就山隆修車廠所扣得之電腦程式列印資料而言：扣案證據中並無DOS系統之程式，或系爭電腦程式著作之原始程式。而車輛檢驗線軟體程式並非僅告訴人能撰寫，不能因應用程式之名稱相同，未經鑑定即當然認為其內容亦相同。

C 就東陽修車廠所扣得之證據資料而言：該檢驗線軟體並非當

<sup>28</sup> 判決全文請參閱司法院網站<http://www.judicial.gov.tw/> (2003/9/29) 台灣台北地方法院八十九年度易字第三四二號判決



初硬碟中原有之程式。

2、法院對前述證據資料的之判斷：

(1) 對於臺北監理處的證據資料部分：

法官係就自己之裁量，只依據告訴人提供之”臺科大鑑定比較表”，其中有針對扣案硬碟中D O S作業系統下之程式進行鑑定比對，而U N I X作業系統之程式因硬碟已毀損而告訴人委託之鑑定人無從鑑定。此外亦只本於告訴人自行製作之”臺科大鑑定比較表解讀”委託證人就此部分於偵查程序中調查並經具結後做出如下證言「...依告訴人提供之臺科大鑑定比較表所示，扣案硬碟中確實存有未經告訴人授權使用之原始程式..」。

(2) 對於東陽修車廠、山隆修車廠的證據資料部分：

法院所依據的仍然是”告訴人所製作之山隆修車廠電腦程式侵害著作權說明資料”，而只調查硬碟內程式列印資料所示之”建檔日期”並且做出「對於該硬碟內有多項程式係”重製”或”修改”自告訴人之系爭電腦程式著作，已據”告訴人”於前述侵害著作權說明資料中詳予說明及列載，縱未經鑑定，亦足認定重製及改作之事實。」的法律上判斷。

(二) 就最高法院九十年台字第五三五號判決<sup>29</sup>而言：

本件判決最高法院引用著作權法的基本觀念，其提到「著作權所保障者，乃著作之「表達方式」(或稱表現形式)，而不及於該著作表達方式所蘊含之思想、程序、製程、系統、操作方法、構想及原理等....換言之著作是否相同，應視其著作之表達方式是否相同為斷。」而對於高等法院認定”電腦程式著作間是否有重製、改作、編輯事實之有無」的

<sup>29</sup> 判決全文請參閱 司法院網站<http://www.judicial.gov.tw/> (2003/9/29)最高法院九十年台字第五三五號判決



判斷方式”，表示否定，亦即高等法院只判斷「原始碼<sup>30</sup>」就可以做出「二個電腦程式係使用相同的表達方式」的論證方式有適用法則不當的錯誤。而；其於發回理由中說明高等法院仍然必須就「二者之表達方式是否同一」的事實重新加以釐清。

### 小結：

就「是否有無交付鑑定的必要性」的爭議點而言，本段引用的第一個判決中，法官只單就”告訴人”所提出之鑑定比較表作為判斷依據，而並未交付鑑定就逕行做出結論。而對於「電腦程式著作間是否有重製、改作事實之有無」的判斷方式的議題上，這二個判決當中的判斷方式所共同顯示出的是：前者法官是依據偵察階段中，證人所做出之證言為依據，並未於審判階段針對被告所提出本文就前段所述及的”基於電腦程式設計的本質”所作的抗辯於以審酌，使其針對證人於”偵查階段所做之證言”在”審判期日中”有詰問<sup>31</sup>之機會；後者判決的最高法院法官雖然認為「只就原始碼來比對程式的表達方式是否相同」的證據判斷方法屬於適用法則不當，但其在判決發回理由中，如果能像本文中所引用的Altai 判決的法官一樣，能夠具體指明下級審法院應為如何之調查事實與適用證據的方式，則就可以在當事人間同時取得一個在訴訟程序上以及著作權法上的利益平衡。

<sup>30</sup> 該二審法院判決理由謂：

「按電腦程式之軟體著作是否構成抄襲，必須比對程式之原始碼……縱使兩個程式著作所表現的螢幕方式雷同，尚不能認為兩者之原始碼即相同……被告販賣之程式是否構成抄襲告訴人之程式著作，必須將二個程式之原始碼加以比對始能決定」（見原判決理由四），似認定電腦程式著作是否構成抄襲（重製），必須比對程式之原始碼始能決定，則依上開說明，原判決難謂無適用法則不當之違誤。

<sup>31</sup> 請參照刑事訴訟法第一百八十四條（證人之隔別訊問與對質）「證人有數人者，應分別訊問之；其未經訊問者，非經許可，不得在場。因發見真實之必要，得命證人與他證人或被告對質，亦得依被告之聲請，命與證人對質。」



## 陸、結論

電腦程式有與一般文字著作不同的性質，依照我國已經通過施行的新刑事訴訟法，法官在具體的訴訟程序當中，尤其是「審判期日的證據調查階段」而言，對於「判斷有無重製、改作電腦程式著作」的證據資料的調查上，不應該逕行自行判斷或者逕行交付第三人鑑定，而應該尊重當事人基於程序主體之地位，適當的賦予被告主張權利之機會，就應當如何就證據來作調查、其調查證據的範圍、調查證據的方法、並調查證據的順序等等事項，均應由當事人本於程序處分權來加以決定，以便在實體事項如著作權權利範圍大小的判斷上，協助法官正確的區分「何者為著作權保護的部分」、「何者非著作權保護的部分」，並進而正確適當的界定原告得依據著作權法據以主張權利的範圍，如此才能同時保障訴訟當事人其「程序利益」以及「實體利益」，並符合著作權法之立法目的。