

Strumenti giuridici per la tutela del software: scegliere quello giusto

di Roberto di Cosmo

Traduzione italiana, a cura di Giuseppe Prencipe, dell'articolo
"Legal Tools to protect Software: Choosing the Right One "
pubblicato sul vol. IV, No. 3, June 2003 della rivista online UPGrade, a cura del CEPIS

Licenza di libera diffusione del documento

Riassunto

In un momento in cui la Comunità Europea sta considerando la possibilità di cambiare la sua politica pubblica in relazione alla protezione giuridica del software, questo articolo esamina i vantaggi dei diversi strumenti di tutela a disposizione. Si presentano delle precise argomentazioni sull'impatto che i diversi strumenti hanno sullo sviluppo del software e sui processi innovativi e si sollecita il lettore a farsi una propria idea.

Parole chiave: diritto d'autore, free software, proprietà intellettuale, software "open source", politica pubblica, brevettazione del software, marchio di fabbrica.

1. Introduzione

Nel settore della produzione del software abbiamo oggi tre strumenti giuridici di tutela del prodotto dalla concorrenza sleale: il marchio di fabbrica, il diritto d'autore e il brevetto. È assai indicativo il fatto che, negli ultimi anni, la legislazione relativa al marchio è stata uniformemente applicata al software senza contestazione alcuna e, da decenni, si fa riferimento alla legislazione sul diritto d'autore con non molta maggiore opposizione (c'è qualche timida obiezione alle modalità recentemente proposte di tutela del copyright, cioè del diritto d'autore, ma non al copyright stesso), al contrario l'applicazione del diritto brevettuale al software negli USA e la sua introduzione in Europa incontrano forte opposizione sia da parte di azionisti di aziende del software che nel campo della ricerca. Come docente universitario europeo che opera nel settore dell'Informatica, presento qui il mio punto di vista¹.

2. Strumenti giuridici di tutela del software: il marchio di fabbrica, il diritto d'autore o il brevetto?

Tutti quelli che oggi intervengono in questo dibattito concordano nel dire che si deve trovare un modo per dare un giusto riconoscimento all'inventore per la sua attività, dal momento che questa dà come conseguenza un bene di valore per la comunità. Un grosso dibattito verte su quale di questi

¹ Invito a leggere sull'argomento anche il notevole contributo del prof. Ullman [4].

tre strumenti, marchio, copyright e brevetto, sia più adeguato, da un lato a compensare in maniera equa l'invenzione, dall'altro a tutelare l'interesse generale e la libera concorrenza (molti imprenditori mettono ufficialmente sullo stesso piano questi due elementi).

Spesso si sentono generiche asserzioni come queste: *"Internet è il futuro, la ricchezza del futuro è la proprietà intellettuale, è necessario tutelare questa proprietà e, pertanto, si deve ricorrere alla brevettazione del software"*. Affermazione quanto mai sbagliata dal momento che, oltre allo strumento del brevetto, ci sono molti altri modi di proteggere la proprietà intellettuale². Infatti, si può acquisire una notevole quota di mercato ricorrendo semplicemente alla legislazione sul marchio di fabbrica, come avviene per la distribuzione di RedHat della Linux³, oppure si riesce a creare, nel campo dei sistemi operativi, un regime di monopolio che dura da decenni solo ricorrendo al diritto d'autore, come ha fatto Microsoft⁴. A questo punto ci si può legittimamente chiedere cosa sta veramente dietro questo improvviso interesse, in un clima di frenetico movimento, alla brevettazione del software, dopo decenni in cui l'innovazione di software di successo è stata garantita dalla tutela della proprietà intellettuale, così com'è prevista dalla legislazione sul marchio di fabbrica e dal diritto d'autore.

3. Diritto d'autore *versus* brevetto.

Innanzitutto, deve essere chiaro che le basi del diritto brevettuale differiscono *profondamente* da quelle del diritto d'autore (per esigenze di chiarezza tralascio nel seguito di far riferimento alla legislazione sul marchio di fabbrica). Non intendo nemmeno entrare in dettagli tecnici sulle differenze tra il diritto d'autore statunitense e quello francese o europeo dal momento che tali differenze sono minori rispetto a quelle relative al diritto brevettuale.

3.1 Che cosa è tutelato e quanta discrezionalità è data all'autore.

Il copyright tutela uno *specifico* codice di programma, *già concretamente scritto*, così come, diciamo, protegge un romanzo specifico e reale come "Il mastino di Baskerville" di Arthur Conan Doyle vietando la riproduzione o la manipolazione senza autorizzazione. Ad esempio, non si può prendere l'opera di Conan Doyle e cambiare alcune righe o sostituire sulla copertina il nome dell'autore con un altro o mettere in vendita il libro come proprio.

Questo è pressoché quanto può fare il diritto d'autore: Conan Doyle non può impedire che si scrivano storie di acuta investigazione poliziesca su casi misteriosi ed intricati, e fortunatamente è così, altrimenti, Poirot e molti altri personaggi che ci affascinano non avrebbero mai visto la luce (a parte il fatto che se il copyright fosse stato così restrittivo Edgar Allan Poe avrebbe in primo luogo potuto impedire a Conan Doyle di scrivere su Sherlock Holmes).

Nel campo della programmazione ciò significa che non ci si può impossessare del programma di qualcun altro (anche se grosse aziende di software proprietario talvolta lo fanno), *ma è ben accetto il fatto che si produca un programma con uguale funzionalità*. Lo ripetiamo: non si può prendere il codice di Office della Microsoft e inglobarlo in uno proprio, ma certamente si è

² Tralascio intenzionalmente di riferire considerazioni come questa: "I brevetti sono necessari per ottenere capitale di rischio"; o l'osservazione, ridicola ma vera: "Una grande azienda d'oltreoceano ha inviato un esperto per spiegarmi perché ho bisogno dei brevetti di software per essere competitivo con loro".

³ Per quelli che ancora non lo sanno si può scaricare gratis dalla Rete in maniera perfettamente legale l'intera copia del CD di RedHat. Il contratto di licenza non soltanto lo permette, ma lo incoraggia.

⁴ Microsoft è veramente l'ultimo arrivato nella brevettazione del software.

liberi, secondo la normativa del diritto d'autore, di scrivere un codice equivalente, o una suite Office migliore, come ha fatto con successo il progetto OpenOffice.

Ciò appare a prima vista molto giusto: si tutela l'autore affinché non venga del tutto scippato di un proprio originale prodotto, ma, allo stesso tempo, si permette la libera competizione in base al merito, facendo in modo che gli autori concorrano per il miglior romanzo o per la migliore suite per ufficio consentendo, così, quella disseminazione delle idee che rende possibile, con generale soddisfazione, la crescita della ricchezza globale.

D'altra parte, il diritto brevettuale è stato introdotto su una base assai diversa da quella sul diritto d'autore: l'idea era di sollecitare gli inventori a rendere pubblico nei vari passaggi tecnici la loro invenzione, invece di tenerla segreta per proteggere la propria attività. In questo modo la si rendeva subito disponibile e si promuoveva così ulteriormente l'innovazione. In cambio di questa totale apertura all'esterno, gli inventori ricevevano un prezioso privilegio: il diritto esclusivo, per un certo periodo di tempo, concesso dallo Stato e garantito dalla legge, di sfruttamento commerciale di ogni possibile realizzazione della loro invenzione (in questo modo li si liberava efficacemente, per un dato periodo di tempo, dall'incombente di dover affrontare la competizione del libero mercato). Ma affinché questo si risolvesse in un buon affare, tutti devono tener fede alla parte del contratto che li riguarda: per ottenere il brevetto l'inventore deve presentare un'invenzione valida industrialmente, non ovvia, ma che abbia caratteristiche di novità, che funzioni veramente e che non sia, ancora una volta, una sorta di chimera del moto perpetuo o la riscoperta dell'acqua calda. Questo perché, a differenza del copyright che tutela gli autori anche relativamente ad una pessima e inutile opera o ai bit di un codice, un brevetto non scatta "automaticamente", ma è concesso da un ufficio brevetti che ha il compito di valutare approfonditamente la richiesta prima di accettarla.

È, allora, il diritto brevettuale adatto a tutelare il software? Secondo i paragrafi 2c e 3 dell'articolo 52 della Convenzione per il brevetto europeo [1] [2], la risposta è un netto no. Però, alcune potenti lobby stanno facendo del loro meglio per cambiare questa situazione dicendo che il "no" non è una risposta giusta e fanno pressione a che la Comunità europea voti la proposta di una direttiva che introduca in Europa la brevettabilità del software[3].

Fortunatamente non abbiamo bisogno di ricorrere ad elucubrazioni filosofiche per capire che cosa significherebbe per le aziende del software un tale cambiamento di direzione della Comunità europea: un paese d'oltreoceano è già da tempo impegnato a seguire la via della brevettazione del software e così possiamo dare uno sguardo a ciò che è successo lì. A guardare i brevetti di software che sono stati concessi, si vede che essi tutelano un metodo astratto per risolvere un problema, non uno specifico pezzo di codice scritto apposta per risolvere quel particolare problema e che fa parte di un programma che può anche non esistere ancora. Nonostante l'enorme numero di argomentazioni diverse e sofisticate che si possono trovare sulla materia, è un dato di fatto che per un programmatore il brevetto è una cosa profondamente diversa dal diritto d'autore. Se si estende il diritto brevettuale anche al software, allora *non si potranno più scrivere programmi* con funzionalità simili a qualcosa che è tutelato dal brevetto di un concorrente. Non importa quanto il tuo programma sia superiore, né che il programma del concorrente sia fatto male o, peggio, non importa se l'abbia poi davvero scritto o se intenda effettivamente scriverlo. Per fare un programma è necessario ottenere la licenza dal titolare del brevetto e questo comporta delle spese. Ma non è solo questione di soldi. Il titolare del brevetto può negare la licenza di utilizzare una tecnologia chiave sulla base che certe persone gli sono antipatiche.

Credo che un esempio sia assai appropriato a chiarire ciò che la tutela attraverso il brevetto può causare: assai recentemente, una grande azienda di software ha reso disponibile in Internet una parte delle specifiche tecniche di un file system di rete a seguito di un'ordinanza giudiziaria, tuttavia

questa azienda ha tentato di far passare in maniera maliziosa alcune clausole con le quali si vietava agli sviluppatori "open source" di scrivere del codice compatibile con quelle specifiche⁵.

Ma davvero avremmo dovuto accettare il concetto che Conan Doyle fosse legalmente autorizzato a dichiarare che nessuna donna potesse scrivere romanzi "compatibili" con i suoi, che trattassero, cioè, di detective che con le loro doti deduttive risolvono casi misteriosi? Ma così non si sarebbe impedito del tutto all'opera di Agatha Christie di venire alla luce?

3.2 Licenza di uccidere

Scrivere software significa, oggi, affrontare decine di problemi "nuovi" eppure simili, combinando una serie di tecniche di base che formano uno "stato dell'arte" in continua evoluzione. Ordinare valori, disporre dati in efficienti strutture di memorie, creare chiare interfacce per l'utente, bilanciare i carichi di lavoro su un set di processori, scambiare dati tra computer remoti, leggere e scrivere da e in differenti formati e così molti altri diversi compiti sono parte del lavoro quotidiano di ogni programmatore e ciò che rende il suo lavoro unico è, assai spesso, non una particolare scelta dell'algoritmo o la struttura dei dati, ma il modo del tutto peculiare con cui egli trasforma l'algoritmo e la struttura dei dati in effettivo codice.

Ciò è confortato dalla nostra quotidiana esperienza nel mondo accademico. Quando affidiamo un'esercitazione agli studenti, accade spesso che, dato uno specifico problema chiaramente formulato, sono pochi, o uno solo, gli algoritmi o le strutture di dati che forniscono una soluzione davvero valida. E però, tra questi, non ci sono due programmi identici (a meno che, naturalmente, qualcuno non abbia violato la legge sul diritto d'autore).

Ora, ogni sistema che utilizzi applicazioni software abbastanza sofisticate ha bisogno di far riferimento a un codice che implementi migliaia di metodi che vanno da idee ingenue ad algoritmi assai complessi. Mentre si può essere d'accordo che algoritmi sofisticati meritino davvero un riconoscimento, perché rappresentano significativi passi in avanti nella scienza, è molto raro, però, il caso che tali avanzamenti siano resi noti in occasione della concessione di un brevetto (il protocollo a chiave pubblica RSA è una notevole eccezione abbastanza rara). La maggior parte delle volte i progressi sono documentati in riviste scientifiche e messi a disposizione di tutti, e ciò è molto ragionevole, dal momento che il più delle volte tali risultati sono ottenuti attraverso finanziamenti pubblici.

D'altra parte, un numero davvero preoccupante di brevetti di software che sono stati dati oltreoceano coprono idee desolatamente ingenue, come un "viaggiatore frettoloso" euristicamente in cerca di un pasto veloce (brevetto #5.249.290) o come le varianti di bozze di lavoro di qualche comitato per la standardizzazione (vedi il ben noto brevetto *Cascading Style Sheet* concesso alla Microsoft) o anche banali metodi di pratica aziendale, come il brevetto di Amazon *one-click buy* ("compra con un clic")⁶.

⁵ Vedi: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnkerb/html/Finalcifs_LicenseAgrmnt_032802.asp>

⁶ Su questo voglio essere chiaro. Amazon ha fatto un meraviglioso lavoro con la vendita di libri online e ha costruito un sistema davvero straordinario per sviluppare la sua azienda, inclusa l'implementazione assai sofisticata di un sistema sicuro che rende possibile il metodo "compra con un clic" alla moltitudine dei clienti che vi ricorrono. Ma mentre l'implementazione è sofisticata ed è giusta che sia protetta, e lo è dal diritto d'autore, l'idea alla base del "compra con un clic" è semplicemente banale.

3.3 Il brevetto per la tutela del software è *speciale*

Il grosso problema è che non c'è spesso modo di bloccare le idee banali e non è possibile in nessun modo assicurarsi l'interoperabilità con protocolli, formati o standard brevettati, se il titolare del brevetto rifiuta di concedere la licenza.

Questo significa che un brevetto abbastanza ampio, abbastanza vago, abbastanza ovvio, è una vera licenza di uccidere i concorrenti. Si consideri il caso di una grande azienda di fronte ad una giovane e dinamica azienda "start-up" che è nata su un'idea veramente forte e che ha il potenziale per diventare un'azienda davvero competitiva in pochi anni. Questa azienda naturalmente può brevettare la sua idea. Ma non potrà mai trasformarla in un prodotto: appena tenterà il rilascio del codice, si vedrà immediatamente che questo codice implementa un'ampia gamma di comuni funzionalità accessorie, e almeno alcune di esse saranno probabilmente protette da qualche banale brevetto in mano della grande azienda.

Di conseguenza, l'unica soluzione per sopravvivere è che la grande azienda rilevi la "start up" e ciò qualche volta può essere soddisfacente per il capitale di rischio che riesce a realizzare un modesto profitto, ma non è mai soddisfacente in termini di competizione e di libero mercato, tantomeno in termini di innovazione, che viene bloccata da tali pratiche.

Questa è probabilmente la ragione per cui solo in un numero limitato di casi si arriva in tribunale: sia che le due parti siano grandi aziende e quindi vanno poi a un qualche accordo di scambio gratuito di licenze dal loro portfolio di brevetti, sia che una delle parti è un pesce piccolo e allora viene generalmente mangiato senza troppi complimenti.

4. Agire ora

Per concludere, un grosso problema che ha a che fare con la tutela del software tramite lo strumento brevettuale è che nelle aziende di software arrivare a una idea brevettabile è, assai spesso, facile, mentre sviluppare del software che utilizzi un'idea che abbia davvero il requisito della novità è abbastanza costoso e complesso. Di solito implica troppe banali invenzioni che, nella maggior parte dei casi, sono state brevettate da qualche tipo sveglio che tiene poi il dito sul grilletto di una potente arma puntata proprio contro la tua azienda. In tale situazione, il tipo sveglio non ha nemmeno bisogno di essere un'azienda di software⁷.

La prossima volta che ti fermi a riflettere su quale inquadramento giuridico sia più adeguato per il software, il diritto d'autore o il brevetto, ricorda che il primo permette di proteggere i faticosi sviluppi della tecnologia e lo fa già da molto tempo, mentre il diritto brevettuale si presta ad abusi che non permettono tali difficili sviluppi attraverso la presunta violazione di qualche oscuro e banale processo corrispondente ad un software dello stesso valore della scoperta dell'acqua calda.

Chiediti allora se è accettabile dare una licenza di uccidere a un pugno di grosse aziende e lasciarle quindi libere di agire tra la vasta comunità degli sviluppatori di software che lavorano nella piccola e media impresa, il bene più prezioso in Europa per costruire la società e il mercato di domani.

Se ti sei convinto che questo potrebbe essere un problema, prenditi qualche minuto, mentre sei ancora in tempo, per far conoscere il tuo pensiero ai tuoi rappresentanti nel Parlamento europeo.

⁷ Gli Stati Uniti hanno dato vita recentemente a un tipo particolare di azienda (non di software) il cui unico vero scopo è quello di accumulare brevetti software e poi mettersi dietro con discrezione ad aziende di software e spingerle a "pagare i diritti di tutela" minacciando di adire le vie legali.

Ringraziamenti

L'autore desidera ringraziare Juliusz Chroboczek per le sue interessanti osservazioni sulla prima bozza di questo articolo.

Riferimenti

- [1] European Patent Convention, article 52, 1978. <<http://www3.european-patent-office.org/legal/epc/e/ar52.html>>.
- [2] Council Directive on the Legal Protection of Computer Programs (91/250/eec), 14 May 1991. <<http://europa.eu.int/eur-lex/en/index.html>>.
- [3] Proposal for a Directive of the European Parliament and of the Council on the Patentability of Computer-implemented Inventions. Official Journal C 151 E, p. 129–131, 25 June 2002.
- [4] J. Ullman. Ordinary skills in the art. Knuth Prize Award Lecture, 2000. <<http://www-db.stanford.edu/~ullman/pub/focs00.html>>.

Roberto Di Cosmo è professore ordinario alla Université de Paris VII (Francia), in cui è responsabile dei corsi postlaurea di Informatica. Ha insegnato per alcuni anni alla École Normale Supérieure di Parigi ed è ora ricercatore all'INRA. I suoi interessi si volgono ai metodi formali, alla logica matematica, alla programmazione funzionale e parallela. Negli ultimi cinque anni si è attivamente impegnato ad analizzare gli aspetti sociologici dell'informatica. <<http://www.dicosmo.org>>, <roberto@dicosmo.org>.

Giuseppe Prencipe, insegnante e traduttore freelance. Si è laureato in Lettere all'Università Cattolica "S. Cuore" di Milano e ha approfondito le tematiche relative alla documentazione, alla biblioteconomia e alla gestione delle conoscenze conseguendo il diploma in Archivistica presso l'Archivio di Stato di Bari e il diploma di perfezionamento in Storia moderna presso l'Università di Urbino. <giuprencipe@tiscali.it>.