SIMSON L. GARFINKEL
RICHARD M. STALLMAN
MITCHELL KAPOR

Perché i brevetti sono dannosi per il software[1]

I brevetti non possono proteggere o rafforzare l'industria del software; la possono solo paralizzare.

Nel settembre del 1990, gli utenti del popolare programma di elaborazione testi XyWrite ricevettero nella corrispondenza un'inaspettata lettera da XyQuest, Inc., l'editore del programma:

> Nel giugno 1987, abbiamo introdotto la funzionalità di correzione automatica ed espansione abbreviazioni in XyWrite III Plus. Senza che ci fosse noto, nel 1984 è stata depositata una richiesta presso l'Ufficio Brevetti per la riferita caratteristica, che veniva in seguito riconosciuta nel 1988. La compagnia detentrice del brevetto ci contattò nel tardo 1989 e ci informò dell'esistenza del loro brevetto.
>
> Abbiamo deciso di modificare XyWrite III Plus cosicché esso non possa essere considerato trasgressore. La nuova versione di XyWrite III Plus (3.56) incorpora due significativi cambiamenti che riguardano la questione: non sarete più in grado di correggere automaticamente gli errori comuni di ortografia premendo la barra spaziatrice dopo la parola errata. Inoltre, per espandere le abbreviazioni memorizzate nel vostro dizionario personale, dovrete premere control-R od un'altra combinazione di tasti definita.

Xyquest é stata imbrigliata da un brevetto software—uno dei più di duemila brevetti su algoritmi per computer e tecniche software che sono stati emessi dall'Ufficio Brevetti e Marchi degli Stati Uniti dalla metà degli anni 80. Il proprietario del brevetto, Productivity Software, ha dato a Xyquest una scelta: ottenere la licenza del brevetto o togliere l'apprezzata funzionalità da XyWrite, il prodotto di punta della XyQuest. Se la XyQuest rifiutava, il ricorso ad un costoso processo per violazione di brevetto sarebbe stato certo.

Qualche alternativa.

XyQuest cercò di ottenere la licenza per il brevetto, dice Jim Adelson, vice presidente per il marketing, ma Productivity Software continuava a cambiare le condizioni. All'inizio Productivity diceva che XyQuest poteva mantenere la caratteristica in alcune versioni di XyWrite, ma non in altre. Poi la compagnia disse che XyQuest poteva usare una parte dell' "invenzione", ma non altre parti. E Productivity Software continuava ad aumentare l'importo del denaro richiesto. Alla fine XyQuest rinunciò e tolse via quella caratteristica.

XyQuest fu fortunata ad avere questa opzione. Altre ditte—inclusi alcuni dei più grandi e redditizi editori di software—sono stati notificati che i loro prodotti di maggiore rilievo violavano dei brevetti. In uno dei casi più pubblicizzati, una compagnia chiamata Refac International—il cui unico obiettivo è quello di acquisire brevetti e muovere causa giudiziaria per le loro violazioni—citò Lotus, Microsoft, Ashton-Tate, ed altri tre produttori di fogli di calcolo, con la rivendicazione che tutti loro avevano violato il brevetto #4.398.249, che spiega nel dettaglio l'ordine in cui ricalcolare i valori in un modello complesso nel quale un parametro del modello cambia. (Refac ha poi rinunciato alle sue rivendicazioni contro tutte le ditte ad eccezione della Lotus, ma solo perché gli avvocati della compagnia prevedevano migliori possibilità di successo se essi avessero fronteggiato un solo avversario.)

Il brevetto #4.398.249 non ha niente a che vedere con i fogli di calcolo in particolare; la tecnica compare anche in alcuni programmi di disegno grafico e di intelligenza artificiale. E l'idea di ricalcolare i valori in un foglio di calcolo secondo l'ordine specificato nel brevetto è così ovvia che probabilmente viene in mente praticamente a chiunque abbia scritto un programma di foglio di calcolo. Ma per l'Ufficio Brevetti lo standard dell'evidenza è estremamente basso; sono stati

[1] Articolo tratto da Issues in Science and Technology, autunno 1991. Titolo originale: Why Patents Are Bad for Software. Tradotto da Michelangelo Altamore, luglio 1999.

registrati brevetti per idee così elementari che sarebbero potute essere risposte a problemi per il primo anno di un corso di programmazione.

In pratica una volta al mese, le reti di computer della nazione sono scosse dalla notiza di un'altro brevetto basato su un concetto fondamentale che è largamente usato. Sebbene l'Ufficio Brevetti non è preposto a registrare le idee, questo è essenzialmente quello che sta facendo con i brevetti software, facendo a pezzi il domino intellettuale della scienza informatica e consegnandone in pratica i piccoli pezzi ad ogni compagnia che presenta una domanda di brevetto. E questa pratica sta devastando l'industria software americana.

Se il Congresso non agisce subito per ridefinire l'applicabilità della legislazione dei brevetti ai programmi per computer, il campo minato legislativo che si trova davanti all'introduzione di nuovi programmi per computer sarà minaccioso—e potenzialmente costoso—a tal punto che le piccole compagnie saranno effettivamente tagliate fuori dal mercato, mentre le grandi e stabili aziende diventeranno talmente immischiate nelle cause che provocheranno un effetto negativo sull'intera industria.

1 Cosa si sta brevettando?

I brevetti sul software non abbracciano l'intero programma; includono, invece, algoritmi e tecniche—le istruzioni che dicono ad un computer come raggiungere un determinato scopo nel programma. Ogni programma è composto da migliaia di istruzioni. Ma laddove l'unica combinazione di algoritmi e tecniche in un programma è considerato un'espressione (come un libro o una canzone) ed è sottoposta alle leggi del copyright, gli stessi algoritmi e tecniche sono trattate come procedure che hanno i requisiti adatti al brevetto.

Le basi giuridiche per questa idoneità sono perloppiù esili. Le leggi statunitensi non consentono agli inventori, per quanto siano brillanti, di brevettare leggi di natura, e in due casi della Corte Suprema (Gottschalk v. Benson, 1972, e Parker v. Flook, 1978) la Corte estende questo principio agli algoritmi per computer e tecniche software. Però nella causa Diamond v. Diehr del 1981, la Corte disse che potevano essere concessi brevetti per processi industriali che erano controllati da certi algoritmi per computer, e l'Ufficio Brevetti sembra avere interpretato questa decisione come il semaforo verde per la brevettabilità di algoritmi e tecniche software in generale.

Adesso si stanno registrando brevetti software ad una velocità allarmante—stando ad alcuni conteggi, ne vengono emessi più di un migliaio ogni anno. Sfortunatamente, la maggior parte delle invenzioni brevettate hanno in sé press'a poco tanta bravura e originalità quanta se ne trova nella ricetta del riso bollito—semplice di per sé ma parte essenziale di piatti complicati. Molti riguardano algoritmi e tecniche davvero elementari e specifiche che vengono usate in un'ampia varietà di programmi. Frequentemente le "invenzioni" menzionate in una applicazione brevettata sono già in uso da altri programmatori al momento in cui viene presentata la domanda per il brevetto.

Quando l'Ufficio Brevetti concede un brevetto su un algoritmo o una tecnica, sta dicendo ai programmatori che essi non devono usare un particolare metodo per risolvere un problema senza il permesso del "proprietario" dell'idea. Per i programmatori, brevettare un algoritmo o una tecnica è come brevettare una serie di note musicali o un giro di accordi, forzando di conseguenza i compositori ad acquistare una "licenza di sequenza musicale".

2 Un sistema in contraddizione

La tradizionale ragione fondamentale, che giustifica l'uso dei brevetti, consiste nel fatto che la protezione delle invenzioni sproni l'innovazione e aiuti la diffusione di informazioni sui progressi della tecnica. Proibendo agli altri di copiare l'invenzione, i brevetti consentono agli inventori di recuperare i loro investimenti nello sviluppo, mentre allo stesso tempo rivelano i dettagli della nuova invenzione al pubblico.

Ma c'è l'evidenza che il sistema brevettuale si stia ritorcendo contro l'industria del software; in realtà, il sistema stesso sembra inadatto alla natura dello sviluppo del software. Gli odierni programmi per computer sono così complessi che essi contengono letteralmente centinaia di algoritmi e tecniche software, ed ognuna viene considerata brevettabile secondo i criteri dell'Ufficio Brevetti. È ragionevole aspettarsi che una software house ottenga una licenza per l'uso di ognuno di quei brevetti, oppure che introduca un prodotto così legalmente rischioso nel mercato? Per rendere le cose ancora più complicate, l'Ufficio Brevetti ha anche riconosciuto i brevetti su combinazioni di algoritmi e tecniche che producono una particolare funzionalità . Per esempio, Apple fu denunciata perché il suo programma HyperCard, secondo l'accusa, violava il brevetto #4.736.308, che include una specifica tecnica la quale, per semplificare il discorso, comporta lo scrolling attraverso un database mostrando le parti selezionate di ogni riga di testo. Separatamente, le funzioni di scrolling e di presentazione dei risultati sono dappertutto nella programmazione di computer, ma sembra che combinarli senza la licenza del detentore del brevetto #4.736.308, adesso, sia illegale.

Un altro problema con il brevetto sul software è l'ammontare del tempo necessario per ottenerlo. Il tempo, dai due ai cinque anni, richiesto per presentare ed ottenere un brevetto è accettabile se la compagnia sta brevettando, diciamo, la formula per il Valium, che non è cambiata in più di vent'anni. Ma nell'industria del software, le compagnie che non rilasciano continuamente le nuove versioni dei loro programmi falliscono. Il loro successo dipende nell'individuare il più velocemente possibile le necessità e svilupparne le soluzioni.

Sfortunatamente, condurre una ricerca per la novità e l'"attivita inventiva" di un brevetto per determinarne i meriti di ammisibilità è un processo lento, che, quando imbriglia lo sviluppo del software, può stroncare l'innovazione alla sua nascita. E poiché le domande di brevetto sono segrete, non c'è veramente modo per i programmatori di computer di assicurarsi che quello che scrivono non violerà qualche brevetto che ancora deve essere rilasciato. A causa di ciò XyQuest "inventò" indipendentemente il suo sistema di correzione automatico degli errori di ortografia e portò il prodotto sul mercato nel tempo intercorso tra il periodo in cui Productivity Software ebbe passato agli atti la sua richiesta ed il momento in cui le fu assegnato il brevetto.

Tali esempi stanno diventando incredibilmente comuni. In un'altro caso, il periodico IEEE Computer, nel giugno 1984, pubblicó un algoritmo altamente efficiente per effettuare compressione di dati; senza che l'editore della rivista o i lettori lo sapessero, gli autori dell'articolo avevano contemporaneamente presentato domanda per un brevetto sulla loro invenzione. Negli anni successivi, per eseguire la cosiddetta "compressione dati LZW" furono scritti e distribuiti ampiamente numerosi programmi. Il sistema di compressione fu anche adottato come standard nazionale e fu proposto come standard internazionale. Poi, nel 1985, l'Ufficio Brevetti assegnò il brevetto #4.558.302 ad uno degli autori dell'articolo. Adesso Unisys, detentore del brevetto, sta richiedendo le royalty per l'uso dell'algoritmo. Sebbene ci sono programmi che incorporano l'algoritmo nel pubblico dominio, usare questi programmi vuol dire rischiare una causa giudiziaria.

Non solo il processo di approvazione del brevetto è lento, ma la ricerca di "prior art"—il criterio che l'Ufficio Brevetti usa per determinare la novità e l'"attività inventiva" determinante per l'ammissibilità del brevetto—è pressoché impossibile da dirigersi nell'ambito del software per computer. Dopo più di 25 anni, l'Ufficio Brevetti non ha sviluppato un sistema di classificazione per algoritmi e tecniche software, e posto che lo facesse nessuno di tali sistemi potrebbe funzionare. Proprio come i matematici che sono qualche volta inconsapevoli del fatto che vengono usati processi mentali essenzialmente uguali in diverse aree della matematica sotto diverse terminologie, così in diversi ambiti della scienza informatica spesso si reinventa lo stesso algoritmo per raggiungere obiettivi differenti. È irragionevole aspettarsi che un esaminatore di brevetti, pressato dalle scadenze, voglia riconoscere tutti questi duplicati. Per esempio, alla IBM è stato rilasciato un brevetto riguardante lo stesso algoritmo di compressione dati di cui la Unisys, a quanto risulterebbe, è proprietaria. L'Ufficio Brevetti probabilmente non era consapevole del fatto che stava concedendo due brevetti per lo stesso algoritmo a causa del fatto che la descrizione dello stesso brevetto era completamente differente, anche se le formule erano matematicamente equivalenti.

La ricerca di "prior art" è complicata dal fatto che la letteratura della scienza informatica è incredibilmente vasta. Essa comprende non solo giornali accademici, ma anche manuali degli utenti, codice sorgente pubblicato e descrizioni in popolari riviste per appassionati di computer. Mentre un'equipe di chimici che lavorano nelle migliori università possono produrre 20 o 30 pagine di materiale pubblicato per ogni anno, un solo programmatore può facilmente produrre cento volte tanto. La situazione diventa anche più complessa nel caso del brevetto su combinazioni di algoritmi e tecniche software. I programmatori pubblicano spesso nuove tecniche, ma nella maggioranza dei casi nessuno di loro pubblica mai nuovi modi di combinare quelle vecchie. Nonostante i singoli algoritmi e tecniche sono stati combinati in molti modi differenti nel passato, non c'è alcun valido modo per stabilire chi storicamente abbia la precedenza.

L'incapacità di cercare esaurientemente nella letteratura per "prior art" è un fatto cruciale, perché a meno che l'esaminatore non trovi del "prior art", lui o lei è obbligato a emettere il brevetto. Come risulta, molti brevetti sono stati assegnati—e in seguito difesi in tribunale—senza che fossero "originali", anche secondo la definizione dell'Ufficio Brevetti. Si trattava semplicemente del fatto che né l'esaminatore del brevetto né i difensori nella causa sapevano del "prior art".

Alcuni membri della comunità del software commerciale stanno adesso proponendo la creazione di un "Istituto del Brevetto Software" (Software Patent Institute) per identificare il "prior art" che esisteva prima del 1980. Ma anche se un tale istituto potrà catalogare ogni scoperta fatta da ogni programmatore degli Stati Uniti, non ha senso dichiarare arbitrariamente che solo prima del 1980 ogni lavoro sul software fosse di pubblico dominio. Inoltre, quale sarebbe l'obiettivo? Per consentire il brevetto di leggi di natura puramente matematica?

3 Dannoso per gli affari

Quantunque i brevetti siano conosciuti in anticipo, gli editori di software in genere non prendono in licenza un algoritmo o una tecnica; al contrario, essi cercano di riscrivere i loro programmi cercando di evitare l'uso di quella particolare procedura che il brevetto descrive. A volte ciò è impossibile; in questo caso le compagnie hanno spesso scelto di evitare completamente di implementare la nuova caratteristica. Sembra chiaro dall'evidenza degli ultimi anni che i brevetti software stanno privando l'adozione di nuove tecnologie, piuttosto che incoraggiarle.

E non sembrano neppure essere favorevoli alle innovazioni. I brevetti software comportano un pericolo particolarmente grave per le piccole aziende, che spesso costituiscono l'avanguardia dello sviluppo del software, sebbene non possano permettersi il costo di ricerche di brevetti o di cause civili. Programmare un nuovo prodotto può costare loro qualche centinaia di migliaia di dollari—il costo di una ricerca di brevetto per ogni singola tecnica e combinazione che il programma usa può essere facilmente uguale o anche maggiore dello stesso costo di produzione. E il costo di una causa civile può ammontare a più di un milione di dollari.

"Io non conosco nessun tipo di cause giudiziarie più costose di quelle per i brevetti", dice R. Duff Thompson, vice presidente e consigliere generale della Wordperfect Corporation. Ma la più grande paura di Thompson è che i brevetti software spazzeranno via i giovani programmatori indipendenti, che fino ad oggi sono stati la fonte di ispirazione per il settore. "Immaginate cosa accade", dice Thompson, "quando qualche ragazzo ventitreenne a cui venga in mente un'idea straordinaria per un programma software, viene bloccato da una lettera con richieste da parte di qualcuno che detiene un brevetto".

Introdotto per aiutare lo scambio di informazioni, l'espansione dei brevetti sul software potrebbe significare invece la fine del software sviluppato nelle università e distribuito gratuitamente—software che è stato il sostegno principale per gli utilizzatori di computer nelle università, enti pubblici e governativi per anni. Molti di questi programmi—l'X Window system, l'editor di testo EMACS, il programma "compress" per comprimere i file e atri—sembrerebbero violare gli esistenti brevetti. I brevetti sul software potrebbero anche significare la fine del software di pubblico dominio, che ha giocato un ruolo determinante nel rendere il costo dei computer sostenibile per le scuole pubbliche. Non c'è ovviamente modo che un autore che distribuisce gratuitamete un programma possa riuscire a pagare le royalty, anche se solo una delle centinaia di tecniche riunite insieme per creare il programma fosse brevettata.

Sono pochi i programmatori e imprenditori che credono che i brevetti siano necessari per la loro professione. Invece, l'impeto di brevettare algoritmi e tecniche giunge da due fonti esterne: i manager di grosse compagnie, che vedono i brevetti come un modo per trionfare sui propri concorrenti senza dover sviluppare prodotti di qualità superiore, e gli avvocati di brevetti, che in essi vedono un potenziale espandersi dei loro affari.

Oggi, la maggioranza delle compagnie brevetta per avere qualcosa con cui negoziare o per difendersi contro le cause di violazioni di brevetti. Gli avvocati sono dell'opinione che brevettare il software rafforza la competitività. Nonostante questo approccio funzioni per le grosse compagnie come Microsoft, Apple e IBM, le compagnie piccole e anche medie non potranno mai essere ad armi pari rispetto a loro. Una nuova ditta futura verrà costretta a pagare qualsiasi prezzo i giganti le vogliano imporle.

4 Copyright e il segreto industriale

Il miglior argomento contro l'opportunità dei brevetti sul software si può trovare nella storia stessa. Lotus, Microsoft, WordPerfect, e Novell sono tutte diventate leader mondiali nel mercato del software per la potenza dei loro prodotti. Nessuna di queste compagnie aveva bisogno di brevetti per assicurarsi risarcimenti fatti nella ricerca o per mantenere la posizione nel mercato. Al contrario, tutti hanno ottenuto la loro posizione dominante nel mercato prima dell'odierna esplosione dei brevetti software. Chiaramente, i brevetti non sono necessari a garantire lo sviluppo dei programmi per computer. E per coloro i quali vogliono un maggiore controllo su ciò che vedono come loro proprietà, l'industria del software ha già adottato due sistemi: il copyright e il segreto industriale.

Oggi, quasi tutti i programmi sono coperti da copyright. Il copyright proibisce all'utente del programma software di fare copie di esso (per esempio, per darlo ad amici) senza il permesso personale o della compagnia che licenzia il programma. Una singola tecnica software può essere implementata in modi differenti per svolgere lavori totalmente differenti; il copyright proibisce solamente di appropriarsi del codice effettivo che un certo programmatore ha scritto.

In generale, si pensa che l'uso di copyright e di brevetti si applichi a campi diversi: il primo all'espressione di idee, e l'altro a procedimenti che ottengono un certo risultato. Fino a qualche anno fa, gli algoritmi per computer erano largamente giudicati non brevettabili. E come nota Brian Kahin, esperto di giurisprudenza dell'università di Harvard, questa è la prima volta nella storia che un'industria nella quale l'uso del copyright si era grandemente stabilizzato è stata improvvisamente soggetta al brevetto.

Infatti, se non ci sarà un'azione giuridica consapevole da parte del Congresso o della Corte Suprema, la norma più fondamentale per un editore di software—se scrivi un programma, ne diventi proprietario—cambierà. La nuova regola è che potresti esserne proprietario—soltanto se il programma è talmente rivoluzionario che non deve nulla ad ogni lavoro precedente. Nessun altro autore in un settore diverso da quello del software è tenuto ad uno standard così elevato.

Il sistema brevettuale degli Stati Uniti fu creato perché i fautori della Costituzione speravano che i brevetti scoraggiassero il segreto industriale. Quando delle tecniche vengono tenute nascoste per vantaggio commerciale, esse potrebbero non essere mai disponibili per l'utilizzo da parte di altri e potrebbero anche perdersi. Ma quantunque il segreto industriale è un problema per il software, come per gli altri campi, esso non è un problema che il brevetto può aiutare a correggere.

Molti degli utili sviluppi nel campo del software consistono in nuove caratteristiche come la correzione automatica e la funzione di espansione automatica in Xywrite III Plus. Poiché non è possibile mantenere segreta una caratteristica agli utenti di un programma, non c'è la possibilità di segreto industriale e quindi non c'è alcun bisogno di scoraggiarlo. Le tecniche usate internamente in un sistema software possono essere mantenute segrete; ma nel passato quelle importanti raramente lo furono. In quel periodo rendere pubbliche le proprie scoperte era normale sia per uno scienziato informatico che lavorava in un'azienda commerciale sia per quello che operava nella comunità accademica. Ancora una volta si vede che, poiché il segreto sulle tecniche non è stato un problema significativo, c'è poco da guadagnare con l'introduzione del sistema dei brevetti per scoraggiarlo.

Il posto in cui il segreto industriale è usato largamente è nel "codice sorgente" dei programmi. Nella programmazione di computer il segreto industriale viene ottentuto distribuendo il programma nel "codice macchina", la traduzione praticamete indecifrabile in un linguaggio di programmazione che il computer può leggere. È estremamente difficile per un altro programmatore scoprire da un programma in linguaggio macchina i procedimenti originali scritti dall'autore del programma. Ma i brevetti software non hanno fatto nulla per limitare questa forma di segreto industriale. Con il detenere il codice sorgente, le compagnie non mantengono il segreto su una tecnica particolare, ma sul modo in cui essi hanno combinato dozzine di tecniche per produrre un sistema completo. Anche le compagnie che dispongono di brevetti sul software distribuiscono i programmi nel solo codice macchina. Quindi, il brevetto del software non riduce significativamente il segreto industriale in nessun modo.

5 Invertire rotta

Molti legislatori presumono che ogni aumento della protezione nella proprietà intellettuale deve essere un vantaggio per chiunque operi nel settore. Come abbiamo cercato di mostrare, questo non è sicuramente il caso della programmazione di computer. Quasi tutti i programmatori vedono i brevetti come una sgradita intrusione, che limita sia la loro capacità di lavorare che la loro libertà di espressione.

A questo punto, sono stati emessi dall'Ufficio Brevetti e Marchi così tanti brevetti che la prospettiva di invalidarli trovando "prior art", uno per volta, è impensabile. Anche se l'Ufficio Brevetti imparasse a comprendere meglio il software in futuro, se non ci sarà una radicale inversione di questo orientamento, gli errori che si stanno commettendo adesso colpiranno l'industria nel prossimo secolo.

L'Ufficio Brevetti e Marchi ha recentemente istituito una commissione consultiva sulla riforma della legislazione sui brevetti che è stata incaricata di esaminare una varietà di argomenti, inclusi i brevetti sul software—o, come preferisce chiamarli, "invenzioni connesse ai programmi di computer". Sfortunatamente, il sottocomitato della commissione non include nessun rappresentante di ditte software che hanno espresso dubbi circa il brevetto del software. Però alla sottocommissione è stato richiesto di tenere in considerazione l'opinione pubblica. Il rapporto finale della commissione non si avrà prima dell'agosto del 1992, quindi c'è ancora tempo per far sentire la propria voce.

Sebbene influenzare l'Ufficio Brevetti può portare qualche beneficio, è probabile che la riforma necessaria possa venire solo attraverso la Corte Suprema o il Congresso. Aspettare la Corte Suprema non serve: nessuno può forzare la Corte Suprema a decidere su un caso rilevante, e non c'è nessuna garanzia che la Corte decida di modificare la pratica dell'Ufficio Brevetti o di fare qualcosa dei brevetti esistenti. Il più efficace modo di agire, quindi, è quello di incoraggiare il Congresso a emendare la legislazione del brevetto per non consentire i brevetti sul software e, se possibile, invalidare quelli che sono già stati riconosciuti. La Sottocommissione alla Camera per la Proprietà Intellettuale e Amministrazione Giudiziaria, presieduta dal deputato William J. Hughes (D-N.J.), dovrebbe cominciare con il fissare le date delle udienze sugli argomenti e convocando una analisi economica patrocinata da Congresso sugli effetti del brevetto software sull'industria.

L'industria del computer è cresciuta sana e forte senza i brevetti. A meno che i proponenti di brevetti sul software riescano a dimostrare che è necessario averli per rafforzare l'industria del software, il Congresso dovrebbe sentirsi giustificato nell'eliminare questa barriera all'innovazione.

6 Letture consigliate

- Brian Kahin, "The Software Patent Crisis," *Technology Review* (April 1990): 53-58.
- Mitchell Kapor, Testimony at Hearings before U.S. House of Representatives, Subcommittee on Courts, Intellectual Property and the Administration of Justice, of the Committee on the Judiciary (March 5, 1990).
- Pamela Samuelson, "Benson Revisited: Should Patent Protection Be Available for Algorithms and Other Computer Program-Related Inventions?" *Emory Law Journal* (Fall 1990): 1025-1154.
- Pamela Samuelson, "Should Program Algorithms Be Patented?" *Communications of the ACM* (August 1990): 23-27.

Simson Garfinkel è capo redattore della rivista *NeXTWORLD* e coautore del libro *Practical UNIX Security.*

Richard M. Stallman è uno dei fondatori della League for Programming Freedom (Cambridge, Massachusetts) e beneficiario del Grace Hopper Award dell'Association for Computing Machinery.

Mitchell Kapor, è fondatore di Lotus Development Corporation e presidente della Electronic Frontier Foundation (Cambridge, Massachusetts).