



Licenze di software libero ed altre licenze libere: codice genetico di beni comuni digitali

Marco Ciurcina
StudioLegale.it
Via San Quintino n. 32, Torino, Italia

Abstract

Questo articolo esplora la storia e descrive le caratteristiche principali delle licenze di software libero e delle altre licenze libere nel tentativo di gettare luce sul perché hanno avuto successo nel promuovere comportamenti individuali convergenti verso la costruzione collettiva di beni comuni digitali.

This article explores the history and describes the main features of free software licenses and other free licenses in an attempt to shed light on the reasons for their success in promoting individual behaviors converging towards the collective construction of digital commons.

Published 28 June 2017

Correspondence should be addressed to Marco Ciurcina, StudioLegale.it, Via San Quintino n. 32, Torino, Italia. Email: ciurcina@studiolegale.it

DigitCult, Scientific Journal on Digital Cultures is an academic journal of international scope, peer-reviewed and open access, aiming to value international research and to present current debate on digital culture, technological innovation and social change. ISSN: 2531-5994. URL: <http://www.digitcult.it>

Copyright rests with the authors. This work is released under a Creative Commons Attribution (IT) Licence, version 3.0. For details please see <http://creativecommons.org/licenses/by/3.0/it/>



Introduzione

Le licenze di software libero sono emerse negli anni '80 come strumento legale per promuovere la costruzione e la diffusione del software libero. I programmi di software libero sono beni comuni digitali perché possono essere utilizzati, modificati e distribuiti da chiunque¹. Più di recente, sono state elaborate altre licenze libere² per costruire beni comuni digitali costituiti da opere creative di tipo diverso dal software. L'articolo descrive alcuni profili legali ed etici nel tentativo di offrire spunti e contribuire all'individuazione delle ragioni che hanno portato allo sviluppo delle licenze libere e di beni comuni digitali costituiti da software ed altre opere.

Il Software Libero

Nella definizione di software libero si legge:

*“L'espressione "software libero" si riferisce alla libertà dell'utente di eseguire, copiare, distribuire, studiare, cambiare e migliorare il software. Più precisamente, esso si riferisce a quattro tipi di libertà per gli utenti del software: Libertà di eseguire il programma, per qualsiasi scopo (libertà 0). Libertà di studiare come funziona il programma e adattarlo alle proprie necessità (libertà 1). L'accesso al codice sorgente ne è un prerequisito. Libertà di ridistribuire copie in modo da aiutare il prossimo (libertà 2). Libertà di migliorare il programma e distribuirne pubblicamente i miglioramenti, in modo tale che tutta la comunità ne tragga beneficio (libertà 3). L'accesso al codice sorgente ne è un prerequisito. Un programma è software libero se l'utente ha tutte queste libertà.”³.
“Il "Software libero" è una questione di libertà, non di prezzo”⁴.*

Non importa dunque se il software viene fornito dietro pagamento di un prezzo: questo non ne cambia la natura.

La disponibilità del codice sorgente, cioè della versione del software che può essere analizzata e modificata da parte degli informatici⁵, è centrale nella nozione di software libero. Per funzionare sui computer il software deve essere tradotto in linguaggio macchina, idoneo a funzionare nei computer, mediante un programma interprete o mediante un programma di compilazione del “codice oggetto” (la versione del software che può essere interpretata dal computer ma che è incomprensibile per i programmatori).

Se si dispone solo della versione in formato oggetto è praticamente impossibile studiare e modificare il software come previsto dalla definizione di software libero.

L'origine del Software Libero

Prima di approfondire la nozione di software libero è opportuno spendere qualche parola sulle origini del fenomeno, andando alle vicende che determinarono, a cavallo degli anni '70 ed '80, l'incontro tra informatica e diritto.

Oggi è naturale pensare al software come oggetto di protezione ai sensi del diritto d'autore (e, in alcuni paesi, del brevetto per invenzione). Ma la storia del software è una storia recente, come anche è recente il dibattito sulla sua tutela giuridica. È negli anni '60 che il software

¹ Nel quadro di questo articolo si adotta questa definizione di bene comune digitale.

² Si definiscono licenze libere quelle secondo le quali sono licenziati beni comuni digitali.

³ Vedi <http://www.gnu.org/philosophy/free-sw.it.html>

⁴ Come si legge in <http://www.gnu.org/philosophy/free-sw.it.html>

⁵ O, come definisce la licenza GNU-GPL, la “forma più indicata dell'opera per effettuare modifiche su di essa”: anche nel caso di programmi realizzati in linguaggio non compilato (per es. javascript, php, python), anche se non viene distribuita una versione in linguaggio macchina del programma, è importante avere comunque accesso al codice nella forma più adatta per realizzare modifiche (onde evitare che si faccia ricorso a pratiche di offuscamento).

muove i primi incerti passi fuori dai laboratori di ricerca. Fino agli anni '70 è considerato un accessorio dei computer (allora costituiti dai costosissimi *mainframe*) e continua a circolare liberamente nelle università secondo gli usi di libera condivisione della conoscenza che caratterizzavano in modo prevalente l'ambiente scientifico ed accademico (almeno a quel tempo). In quegli anni si diffonde l'idea che il software possa avere valore economico autonomo rispetto a quello dell'hardware sul quale viene installato e, negli Stati Uniti (il paese in cui l'informatica è più sviluppata), si discute in modo acceso sull'opportunità di tutelare giuridicamente il software⁶.

In astratto, i modelli di tutela possibili per il software erano due: diritto d'autore e brevetto. Dopo lungo dibattito si optò per il modello più elastico e meno burocratico del diritto d'autore: copia, modifica e distribuzione del software devono essere autorizzate dall'autore ma quest'ultimo non è costretto a ricorrere preventivamente ad alcun procedimento amministrativo per ottenere l'accertamento del suo diritto. È interessante rilevare come questa scelta è stata il frutto delle pressioni provenienti dall'industria dell'hardware convinta, allora, che la tutela offerta al software dal diritto d'autore fosse la soluzione migliore. Infatti, si diceva, la tutela del diritto d'autore opera automaticamente, a differenza di quella brevettuale, ed è flessibile, consentendo di tutelare, oltre alla versione originaria, anche le modifiche del software.

Nella prassi, lo *US Copyright Office* accettava domande di registrazione di software sin dal 1964, ma per parecchi anni ancora sia le norme che la giurisprudenza non offrirono appigli sicuri per argomentare l'estensione del diritto d'autore al software. Questa situazione d'incertezza termina nel 1976 con una riforma del diritto d'autore che apre alla tutela del software ed il dibattito si chiude definitivamente nel 1980, quando viene emanato il *Computer Software Copyright Act*⁷. Ma negli stessi anni le corti statunitensi si pronunciano per l'applicabilità del diritto di brevetto per invenzione al software: nel 1981 la Corte Suprema degli Stati Uniti, con il caso *Diamond v. Diehr*, statuisce per la prima volta ammettendo la brevettabilità del software⁸.

Intanto si diffonde il personal computer e l'informatica a basso costo: il crollo dei costi dell'hardware apre la strada al mercato del software commerciale. In quegli stessi anni negli USA accadono altre vicende che concorrono ad incidere profondamente sul modo in cui si lavora nei laboratori informatici delle università americane. Nel 1980 viene emanato un atto normativo, il *Bayh Doyle Act*, che consente alle università di fare ricerca congiuntamente ad imprese commerciali e di privatizzare i risultati della ricerca. Da tempo, per far fronte alle proprie esigenze di gestione dei servizi di telecomunicazioni, AT&T, nota società americana operante nel settore delle telecomunicazioni, aveva realizzato il sistema operativo *UNIX*. Non potendolo commercializzare, lo rendeva liberamente disponibile alle università statunitensi: *UNIX* era il sistema operativo utilizzato nei laboratori informatici di queste. Negli stessi anni l'impresa commerciale AT&T si scinde in diverse società, cosicché per i soggetti che risultano dalla scissione viene meno il divieto di svolgere attività d'impresa in settori diversi da quello delle telecomunicazioni: fino ad allora, infatti, l'autorità antitrust statunitense aveva consentito che AT&T fornisse servizi di telecomunicazione in condizione di monopolio a certe condizioni, tra le quali il divieto di realizzare attività commerciali in mercati diversi. Quindi, dal 1982, a seguito del frazionamento di AT&T, *UNIX* viene distribuito con licenza commerciale. Nei primi anni '80, mentre prende forma il modello del software proprietario, le pratiche di condivisione e libera circolazione dei software, proprie dei laboratori informatici delle università americane, sono sotto assedio.

⁶ Su questo processo ha inciso anche lo sviluppo e la diffusione del linguaggio di programmazione C: il linguaggio originariamente sviluppato (tra il 1969 e il 1973) per la stesura del sistema operativo UNIX che si è progressivamente imposto come linguaggio di riferimento per la scrittura di software su gran parte delle piattaforme hardware (vedi <https://www.bell-labs.com/usr/dmr/www/chist.html> – consultato il 15 giugno 2017).

⁷ A questa fanno seguito, negli anni successivi, numerose legislazioni in vari paesi nel mondo. Nell'UE il 14 maggio 1991 viene adottata la Direttiva n. 91/250/CEE relativa alla protezione giuridica dei programmi per elaboratore (che è stata successivamente sostituita dalla Direttiva n. 2009/24/CE), recepita con legge in tutti i paesi dell'Unione, attuata in Italia con il D. Lgs. n. 518 del 29 dicembre 1992 in forza del quale si è riconosciuta piena tutela di diritto d'autore al software.

⁸ Vedi <https://www.law.cornell.edu/supremecourt/text/450/175>

Le Licenze di Software Libero

In questo contesto, una parte della comunità degli sviluppatori rifiuta di conformarsi al nuovo paradigma del software proprietario. Nel 1983 Richard M. Stallman fonda il Progetto *GNU* con l'intenzione di creare un sistema operativo completamente libero alternativo al sistema *UNIX*, sottratto alla libera disponibilità degli utenti (infatti l'acronimo ricorsivo *GNU* sta per "*GNU's not UNIX*": "*GNU* non è *UNIX*"). Molti sviluppatori si uniscono allo sforzo di Stallman e, grazie alle potenzialità della rete Internet, anch'essa agli albori ed in sviluppo rapido e tumultuoso, il Progetto *GNU* cresce rapidamente. Ma l'estensione del diritto d'autore al software riserva al titolare di questo i diritti di pubblicazione e di utilizzazione economica del software.

Il diritto d'autore opera di default: anche se il titolare del diritto non fa nulla (non registra il software e non lo accompagna con informative o licenze), il suo software è protetto dal diritto d'autore e, senza la sua autorizzazione, non può essere utilizzato. Si risolve il problema utilizzando lo strumento giuridico della licenza: il movimento del software libero si organizza intorno alle licenze di software libero. Perché un programma sia software libero è quindi sufficiente che il titolare del diritto lo pubblichi applicandovi una idonea licenza: una licenza di software libero.

Nel 1989 Richard Stallman scrive la prima versione della licenza GNU General Public License (detta anche GNU-GPL)⁹, unificando licenze simili utilizzate per le versioni precedenti dei suoi programmi. La licenza GNU-GPL viene quindi utilizzata per i programmi che fanno parte del progetto GNU; oggi è utilizzata per un grandissimo numero di progetti e sta al cuore del movimento per il software libero. La grande diffusione di questa licenza è in parte dovuta a ragioni storiche (è la licenza realizzata da Richard Stallman, fondatore del movimento del software libero) ma anche a ragioni pratiche: l'ingegneria della licenza ne ha favorito la diffusione. La GNU-GPL, infatti, stabilisce che è legittimo modificare e redistribuire il software licenziato secondo questa licenza a condizione che la versione modificata sia a sua volta licenziata secondo i termini della stessa licenza o di altra compatibile. È il meccanismo "*copyleft*"¹⁰ che ha dimostrato di avere grande forza attrattiva e che ha favorito la diffusione del software libero stesso¹¹.

Per dirla in breve, le licenze "*copyleft*" incentivano la condivisione: chi vuole modificare il software e distribuirlo (o, a volte, farlo utilizzare da remoto) può farlo a condizione che a sua volta conceda ai suoi utenti le stesse libertà che gli sono state concesse¹². Un *hacking*¹³ del diritto (per usare un'espressione comune tra gli informatici) che innesca un meccanismo di diffusione virtuosa del software libero proteggendo le libertà degli utenti.

Quando nel 1992 Linus Torvalds, giovane studente finlandese, rende disponibile come software libero il *kernel Linux* (l'elemento del sistema operativo che gestisce le interazioni tra le diverse parti del computer: tastiera, schermo, cpu, memoria RAM, ecc.), il nuovo sistema operativo *GNU-Linux* è completo. Indubbiamente la realizzazione del *kernel Linux* è stato un evento molto importante nella diffusione del software libero. Forse è anche per questo che oggi, quando leggiamo il termine *Linux*, dobbiamo domandarci se chi lo usa intende riferirsi veramente al *kernel* del sistema operativo o, con una sineddoche d'uso comune, all'intero sistema operativo *GNU-Linux* del quale il *kernel* è solo una parte.

Dopo la realizzazione del *kernel Linux*, l'industria informatica inizia a percepire il potenziale del software libero. Nel 1998 un gruppo di appassionati fonda la *Open Source Initiative*¹⁴ con lo scopo dichiarato di evitare l'enfasi sugli aspetti etici del software libero che, a loro dire, ostacola la comprensione e l'uso del software libero da parte dell'industria informatica. Utilizzano quindi

⁹ Alla quale è seguita una seconda versione del 1991 ed una terza del 2007.

¹⁰ La parola "*copyleft*" esprime un gioco di parole intraducibile in italiano. In inglese, *right* può significare "destra" e "diritto" mentre *left* può significare "sinistra" e "permesso". Quindi "*copyleft*" è il contrario di *copyright* e si potrebbe tradurre "permesso d'autore".

¹¹ Il meccanismo del *copyleft* non è parte necessaria della nozione di software libero. Sono infatti licenze di software libero anche licenze che non sono *copyleft*, come per esempio la cosiddetta licenza BSD modificata (<http://www.xfree86.org/3.3.6/COPYRIGHT2.html#5>).

¹² Vedi <https://www.gnu.org/licenses/copyleft.en.html>

¹³ Contrariamente a quanto si pensa, *hacker* non è chi si introduce maliziosamente nei sistemi informatici di terzi, ma chi trova una soluzione creativa per risolvere un problema. È vero che, per fare intrusioni informatiche, spesso, si devono trovare soluzioni creative a problemi informatici complessi (e quindi si è *hacker*), ma l'espressione giusta per definire chi fa intrusioni informatiche nei sistemi altrui è *cracker*.

¹⁴ Vedi <http://www.opensource.org>

l'espressione "open source" ed adottano la *Open Source Definition*¹⁵, che però, nella sostanza, riproduce con formulazione diversa la definizione di software libero. L'espressione *Open Source* enfatizza il requisito dell'accesso al software in formato anche sorgente: l'OSI mette in sordina gli aspetti etici e punta l'attenzione sul modello di sviluppo del software libero / *open source*.

Analogamente a quanto fa la *Free Software Foundation* (fondata da Richard Stallmann) con riferimento alla definizione di software libero¹⁶, anche l'*Open Source Initiative* tiene un elenco di licenze alle quali riconosce la conformità con la *Open Source Definition*¹⁷, ma se si comparano i due elenchi si rileva che sono sostanzialmente coincidenti e le poche differenze dipendono da questioni di dettaglio più che di sostanza¹⁸.

Oggi esistono molte licenze di software libero, anche se le più utilizzate sono relativamente poche: le 10 licenze più diffuse sono adottate da oltre il 90% dei progetti di software libero¹⁹. L'aspetto più rilevante delle licenze di software libero è costituito dalla presenza o meno in esse di una clausola *copyleft*. La clausola "copyleft" non è uguale in tutte le licenze di software libero e quindi produce effetti diversi a seconda della sua formulazione. Si usa quindi classificare le licenze di software libero in base al modo in cui funziona, in una specifica licenza, la caratteristica "copyleft". Ci sono, innanzitutto, licenze "non-copyleft", cioè licenze (come le licenze BSD²⁰, MIT²¹ e Apache²²) che non contengono una clausola "copyleft" e quindi non hanno effetto "copyleft": chi distribuisce un programma licenziato con una licenza "non-copyleft" non è tenuto a distribuirlo secondo i termini della stessa licenza. Vi sono poi le licenze cosiddette "strong copyleft": sono licenze che contengono clausole "copyleft" che estendono i loro effetti a tutte le opere derivate, comprese le librerie che, quando si esegue il programma licenziato con licenza "strong copyleft", sono collegate dinamicamente a questo²³. Le licenze che, invece, circoscrivono in modo più o meno ampio la portata della clausola "copyleft", permettendo quindi di applicare licenze diverse ad alcune opere derivate, sono dette licenze "weak copyleft"; tra le quali si ricordano la GNU Lesser General Public License (GNU-LGPL)²⁴ e la Mozilla Public License (MPL). Vi sono poi alcune licenze, come per esempio la GNU Affero General Public License (GNU-AGPL)²⁵ e l'European Union Public License (EUPL)²⁶ che impongono di rendere disponibile il codice sorgente del programma anche agli utenti che lo utilizzano da remoto, collegandosi al *server* presso il quale il software è fatto funzionare come servizio (cd. SaaS): queste licenze sono dette "Network copyleft".

In alcuni paesi, al software si può applicare anche il diritto di brevetto per invenzione che attribuisce al titolare il diritto esclusivo di attuare l'invenzione e di trarne profitto.

Chi usa o distribuisce del software libero non può solo per questo escludere che quel software attui un'invenzione tutelata da un brevetto. L'uso e la circolazione del software libero sono quindi interferiti anche dal diritto di brevetto per invenzione.

In alcune licenze di software libero si adottano diverse tecniche per limitare l'interferenza dei brevetti con il software libero e scoraggiare chi voglia impedire l'uso e la circolazione del software libero facendo valere un proprio brevetto per invenzione. Per esempio, alcune licenze prevedono che chi contribuisce al software e/o chi lo distribuisce (a seconda dei casi) concede licenza dei propri (eventuali) diritti di brevetto.

¹⁵ Vedi http://www.opensource.org/docs/definition_plain.php

¹⁶ Vedi <http://www.gnu.org/licenses/license-list.it.html>

¹⁷ Vedi <http://opensource.org/licenses/>

¹⁸ Vedi https://en.wikipedia.org/wiki/Comparison_of_free_and_open-source_software_licenses

¹⁹ Vedi <http://www.blackducksoftware.com/oss/licenses#top20>

²⁰ Ve ne sono diverse versioni, l'ultima delle quali è quella adottata per il progetto FreeBSD (vedi <https://www.freebsd.org/copyright/freebsd-license.html>).

²¹ Vedi <https://mit-license.org/>

²² Per la più recente versione 2.0 vedi <https://www.apache.org/licenses/LICENSE-2.0>

²³ Va dato conto del fatto che sull'estensione dell'effetto copyleft delle licenze strong copyleft vi sono posizioni contrastanti (vedi <http://www.ifosslr.org/public/LinkingDocument.odt> citato in Bain, 2010)

²⁴ Per la più recente versione 3.0 vedi <https://www.gnu.org/licenses/lgpl-3.0.en.html>

²⁵ Per la più recente versione 1.2 vedi https://joinup.ec.europa.eu/community/eupl/og_page/eupl-text-11-12

²⁶ Per la più recente versione 3.0 vedi <https://www.gnu.org/licenses/agpl-3.0.html>

Software Libero ed Etica

Oggi il sistema del software libero è costituito da un grande numero di programmi²⁷ e dall'insieme di relazioni che si costruiscono con questi programmi tra un grande numero di persone (sviluppatori ed utenti), imprese ed enti pubblici o enti *no profit*.

Le persone che sviluppano e usano software libero possono farlo a titolo personale o nell'interesse di imprese o enti per i quali operano.

Se è irragionevole attendersi che le imprese scelgano di sviluppare o usare software libero motivate da ragioni etiche, lo stesso non può escludersi per le persone che agiscono a titolo personale o (assumendo che lo scopo di enti *no profit* possa coincidere con istanze etiche) per gli enti *no profit*.

È un fatto che il software libero nasce, nei primi anni '80, come reazione all'emergere del nuovo paradigma del software proprietario ed in base ad istanze etiche.

Dice Richard Stallman: *“Il mio lavoro sul software libero è motivato da un obiettivo ideale: diffondere la libertà e la cooperazione. Voglio incoraggiare la diffusione del software libero sostituendolo al software proprietario che proibisce la cooperazione, e quindi rendere la nostra società migliore.”*²⁸.

Il software libero non è una semplice alternativa al software proprietario: il software libero, a differenza del software proprietario, è un bene comune digitale a disposizione di tutti. Si può dire che il software sta alla emergente società dell'informazione e della conoscenza come l'acqua²⁹ sta alla società agricola e pre-industriale: una fondamentale *res communis omnium* che va incentivata e tutelata per garantire il florido sviluppo della società.

Il software libero concretizza un ideale sociale e culturale condiviso da una parte importante degli sviluppatori che lo realizzano e degli utenti che lo usano: è lo strumento per eliminare ogni barriera d'accesso alle risorse informatiche per tutti gli esseri umani.

Ma è un fatto che a partire da quell'originaria spinta etica si sono oggi organizzati sistemi socio-tecnologici ai quali partecipano soggetti animati da motivazioni molto diverse da quelle che hanno dato vita al suo “disegno” originario.

Nel prodursi di questo risultato hanno certamente giocato un ruolo fondamentale le licenze di software libero. Intorno ad esse si sono organizzati numerosi progetti di sviluppo di programmi (a volte istituzionalizzati in enti, altre volte no) che possono interagire e scambiare dati, funzioni ed informazioni, a volte in modo esplicito, altre volte no.

È quindi legittimo ipotizzare che le licenze di software libero abbiano funzionato come mezzi di comunicazione idonei a favorire comportamenti stigmergici: le licenze di software libero sono state il codice generativo/genetico di sistemi socio-tecnologici che si sono auto-organizzati intorno ad esse e che, complessivamente presi, sono privi di un coordinamento centralizzato (Elliot, 2006).

Perché nel software libero si è prodotto questo effetto?

Certamente la risposta a questa domanda è un terreno di riflessione che richiede valutazioni diverse da quelle possibili da una prospettiva giuridico-legale.

Da questa prospettiva è però utile osservare che le licenze di software libero sono state efficienti nel risolvere i problemi che tipicamente si gestiscono mediante atti giuridici (norme, contratti, ecc.): eliminano incertezza, minimizzano i costi di transazione e riallocano il rischio.

Le licenze di software libero sono creazioni conosciute e riconosciute nelle comunità di sviluppatori ed utenti di software libero: il fatto che un programma sia disponibile secondo i termini d'una certa licenza rende semplice per l'utente identificare i propri diritti ed obblighi.

L'uso di una licenza di software libero, invece del ricorso ad una licenza redatta *ad hoc*, riduce di ordini di grandezza i costi connessi con la redazione della licenza stessa e aumenta la probabilità che si diffondano pratiche interpretative e giudiziali uniformi.

Infine, se un programma è disponibile secondo i termini d'una licenza di software libero, l'utente può legittimamente contare sul fatto che chi lo distribuisce secondo i termini di quella licenza non abbia consapevolmente incluso codice in violazione del diritto di terzi.

In una parola, le licenze di software libero sono efficienti nel produrre fiducia tra i soggetti che partecipano al sistema socio-tecnologico che si costruisce intorno ad esse.

²⁷ Sul sito <https://www.openhub.net/explore/projects> se ne contano più di 650.000.

²⁸ Da <https://www.fsf.org/licensing/essays/pragmatic.html>

²⁹ L'immagine suggestiva non deve far perdere di vista le differenze tra l'acqua (che, anche se è un bene pubblico, è un bene materiale e quindi ad uso esclusivo) ed il software che, in quanto bene immateriale, può essere utilizzato da chiunque senza che l'uso di uno limiti l'uso degli altri.

È legittimo ipotizzare che dall'efficienza tecnico-legale delle licenze di software libero siano derivati effetti di rilievo sociale, economico e di altro tipo.

È però utile qualche ulteriore considerazione.

Se pare ragionevole ritenere che le clausole *copyleft* siano desiderabili per le comunità di persone particolarmente motivate dallo scopo etico di proteggere le libertà degli utenti e favorire la condivisione, questi obiettivi non sono condivisi dalle comunità di sviluppo in tutte le circostanze. L'interesse dei soggetti interessati ad evitare l'effetto *copyleft* ha portato alla redazione di licenze *weak copyleft* e licenze *non-copyleft* con caratteristiche varie.

Nel tempo, si è potuto verificare che le licenze di software libero non sono state sempre efficienti nel risolvere tutti i problemi legali che conseguono all'uso di software libero: quando questo è successo, si sono però cercate soluzioni ed adattamenti che, complessivamente parlando, hanno garantito la crescita del sistema socio-tecnologico del software libero.

Per esempio, le licenze di software libero, pur adottando alcune tecniche per questo scopo³⁰, sono oggettivamente inidonee a risolvere radicalmente il problema dell'interferenza del diritto brevettuale.

Per questo, da alcuni anni è stato costituito un *pool* brevettuale al quale partecipano i principali attori del settore³¹ che garantisce a tutti gli utenti del *kernel Linux* la licenza dei brevetti detenuti da tutti i membri del *pool*.

Vi sono poi altri profili giuridici che possono essere coinvolti dalla circolazione del software libero (come per esempio il diritto dei marchi³², il diritto sulle misure tecnologiche di protezione o il diritto sulle informazioni segrete). In alcuni casi, la gestione di questi profili giuridici ha trovato soluzione all'interno delle licenze di software libero e/o nel lavoro d'interpretazione delle loro disposizioni. Altre volte ha trovato soluzione nell'adozione di pratiche comunitarie e/o nuovi atti giuridici. Per esempio:

- per gestire la tematica del diritto di marchio, sono state create apposite politiche di marchio che si affiancano alle licenze di software libero per specifici progetti³³;
- è stato elaborato il formato Software Package Data Exchange (SPDX) che consente di documentare in modo standardizzato (e quindi identificare in modo automatico) le informazioni sulle licenze che si applicano al software libero³⁴;
- recentemente, è stata intrapresa un'iniziativa³⁵ con lo scopo di offrire uno standard di riferimento per consentire a chi distribuisce software libero di svolgere attività di verifica del rispetto delle previsioni delle licenze di software libero e quindi migliorare le condotte complessive dei vari attori.

In altri casi, le comunità di sviluppatori e utenti hanno adottato, selezionato e raffinato tecnologie idonee a massimizzare libertà e collaborazione: sistemi di gestione delle versioni del software, sistemi di segnalazione dei banchi, formati aperti, ecc.

Dalle Licenze di Software Libero alle Licenze Libere

Il modello del software libero ha ispirato tentativi di riprodurre le dinamiche anche in altri settori d'attività umana ed ha portato alla creazione di licenze per opere diverse dal software (giornali, libri, musica, video, banche di dati, disegni di schede elettroniche, ecc.).

Nel passaggio verso nuovi ambiti d'applicazione hanno svolto un ruolo fondamentale le *Creative Commons Public Licenses* realizzate dalla *Creative Commons Corporation*, un ente *no*

³⁰Oltre alla licenza (espressa o implicita) dei diritti di brevetto per invenzione citata nel capitolo precedente, alcune licenze prevedono ulteriori tecniche contrattuali d'una certa efficacia. Per esempio, la licenza MIT, prevede una clausola di scioglimento per rappresaglia nel caso in cui l'utente faccia valere dei diritti di brevetto; oppure, la licenza GPLv3 prevede per l'utente l'impegno ad astenersi dal partecipare ad accordi di co-desistenza dall'azionare diritti di brevetto.

³¹Ci si riferisce all'iniziativa *Open Invention Network* (vedi <https://www.openinventionnetwork.com/>).

³²Se il software libero è distribuito con un marchio, può essere necessario rimuovere lo stesso marchio prima di usarlo nell'attività economica.

³³Si veda per esempio la trademark policy della Mozilla Foundation (vedi <https://www.mozilla.org/en-US/foundation/trademarks/policy/>).

³⁴Vedi <https://spdx.org/>

³⁵Ci si riferisce al progetto OpenChain promosso dalla Linux Foundation (vedi <https://www.openchainproject.org/>).

profit statunitense (anche se non sono state le prime licenze standard pensate per opere diverse dal software³⁶ e, dopo di esse, ne sono state create altre).

Le *Creative Commons Public Licenses* sono 6 licenze di diritto d'autore modulari, formate dalla combinazione di quattro opzioni:

- Attribuzione: si deve riconoscere credito all'autore dell'opera;
- Non commerciale: l'opera non può essere usata a fini di lucro o commerciali;
- Non opere derivate: non è consentita la diffusione di opere che modificano o trasformano l'opera;
- Condividi allo stesso modo: eventuali opere che modificano l'opera concessa in licenza devono essere diffuse secondo i termini della licenza che si applica all'opera concessa in licenza.

L'opzione Attribuzione è obbligatoria dalla versione 2.0 delle *Creative Commons Public Licenses* (la versione più recente è la 4.0).

Le opzioni Non opere derivate e Condividi allo stesso modo sono logicamente incompatibili e non possono coesistere in quanto incidono (in modo diverso) sulla stessa facoltà di diffondere opere che modificano quella originariamente concessa in licenza.

La combinazione delle 4 opzioni rende quindi possibili 6 modelli di licenze:

- Attribuzione,
- Attribuzione - Non opere derivate,
- Attribuzione - Condividi allo stesso modo,
- Attribuzione - Non commerciale,
- Attribuzione - Non commerciale - Non opere derivate,
- Attribuzione - Non commerciale – Condividi allo stesso modo.

Le *Creative Commons Public Licenses* sono pensate per offrire all'autore la possibilità di scegliere il modello di licenza più adatto ai propri scopi. Ma, da una parte, l'opzione Non commerciale sottrae agli utenti la libertà di usare l'opera per scopi commerciali e, dall'altra, l'opzione Non opere derivate sottrae all'utente la libertà di modificare l'opera, divergendo quindi dal modello del software libero. Questa differenza è stata evidenziata dai promotori della definizione di Opera Culturale Libera³⁷, secondo la quale sono licenze di opere culturali libere solo due delle *Creative Commons Public Licenses*: la licenza CC Attribuzione e la licenza CC Attribuzione – Condividi allo stesso modo. *Creative Commons Corporation* ha anche realizzato CC0³⁸, una dichiarazione di rinuncia ai diritti d'autore sull'opera.

Le licenze CC Attribuzione, CC Attribuzione Condividi allo stesso e CC0 si sono dimostrate utili alla generazione di beni comuni digitali; per esempio, la licenza CC Attribuzione Condividi allo stesso modo è attualmente utilizzata in Wikipedia³⁹. Negli ultimi anni si sono intensificate le attività di creazione di beni comuni digitali che consistono in banche di dati, anche per la scelta operata da diversi governi di pubblicare banche di dati secondo i termini di licenze libere. Le licenze CC Attribuzione, CC Attribuzione Condividi allo stesso e CC0 sono state utilizzate per le banche di dati, ma si sono create anche nuove licenze libere pensate specificamente per queste, come per esempio le licenze realizzate dalla *Open Data Commons*⁴⁰, tra le quali è particolarmente importante la licenza *Open Data Commons Open Database License*⁴¹, utilizzata nel progetto *Open Street Map*⁴².

Più di recente, si stanno intensificando i tentativi di creare beni comuni digitali connessi alla produzione di oggetti materiali. È utile suddividere questo ambito in due sottocategorie: da una parte, le schede elettroniche, dall'altra gli altri oggetti materiali. La distinzione è utile per ragioni storiche: il movimento per la costruzione di beni comuni digitali relativi alle schede elettroniche (anche per la maggiore vicinanza culturale delle comunità che vi partecipano con quelle degli

³⁶ Si pensi, per esempio alla Gnu Free Documentation License adottata dalla Free Software Foundation per la documentazione relativa al progetto GNU (vedi <https://www.gnu.org/licenses/fdl.html>).

³⁷ Vedi <http://freedomdefined.org/Definition>

³⁸ Vedi <https://creativecommons.org/publicdomain/>

³⁹ Vedi <https://www.wikipedia.org/>

⁴⁰ Vedi <https://opendatacommons.org/>

⁴¹ Vedi <https://opendatacommons.org/licenses/odbl/>

⁴² Vedi <http://www.openstreetmap.org/copyright>

sviluppatori di software libero), è molto più risalente nel tempo, mentre l'interesse per la realizzazione di beni comuni digitali relativi alla produzione di oggetti materiali di tipo diverso si è sviluppato solo negli ultimi anni, dopo la diffusione delle stampanti 3D. La distinzione è utile anche per ragioni pratiche: i quadri normativi che insistono sulle due classi di creazioni sono diversi. Le schede elettroniche possono costituire l'oggetto del diritto sulle topografie di prodotti a semiconduttori; sugli altri oggetti materiali possono invece insistere diritti relativi alla forma dell'oggetto (come il diritto su modelli e disegni).

Le comunità che operano per la costruzione di beni comuni digitali connessi alla produzione di oggetti materiali devono quindi gestire problemi diversi da quelli risolti dalle comunità che sviluppano software libero. Innanzitutto, devono verificare che la licenza libera che scelgano di utilizzare gestisca in modo corretto i quadri normativi coinvolti con la tipologia di creazione che intendono realizzare. Questo problema si è storicamente posto con riferimento alle schede elettroniche: usando la licenza GPL, grazie alla formulazione di questa, si licenziano anche i diritti sulla topografia dei prodotti a semiconduttori, ma non tutte le licenze libere fanno altrettanto.

Le comunità che licenziano disegni di schede elettroniche si sono anche poste il problema di garantire che chi acquista una scheda possa avere accesso al disegno della stessa scheda. Per ottenere questo effetto, sono state realizzate due nuove licenze libere, appositamente pensate per le schede elettroniche: la CERN Open Hardware Licence⁴³ e la TAPR Open Hardware License⁴⁴.

La gestione dei diritti su disegni e modelli (che eventualmente insistano sui disegni di altri oggetti materiali) potrebbe invece richiedere la creazione di nuove licenze o di adattare licenze già esistenti (Margoni, 2013, 225-248). Ma una strategia efficace di costruzione di beni comuni digitali connessi al disegno di oggetti materiali potrebbe richiedere di tenere conto di ulteriori aspetti. Il passaggio dal disegno digitale di un oggetto materiale alla sua produzione impone di confrontarsi con ulteriori quadri normativi, come per esempio quelli che impongono il rispetto di specifici standard di sicurezza o che prevedono l'ottenimento di certificazioni. Se si tiene conto dei quadri normativi che interferiscono con la produzione di un oggetto materiale sin dalla fase di disegno, si possono ridurre i costi di produzione dello stesso oggetto.

Ci si può infine domandare se nella creazione di beni comuni digitali non sia opportuno tenere conto delle norme sulla tutela dei dati personali, in particolare, di alcune novità introdotte dal Regolamento (UE) 2016/679 del Parlamento Europeo e del Consiglio del 27 aprile 2016 relativo alla protezione delle persone fisiche con riguardo al trattamento dei dati personali, nonché alla libera circolazione di tali dati e che abroga la direttiva 95/46/CE (Regolamento generale sulla protezione dei dati) che avrà effetto a partire dal 25 maggio 2018⁴⁵.

Conclusioni

Le licenze di software libero, insieme a pratiche coerenti con i valori delle comunità del software libero, raffinate nel tempo dalle persone ed enti che sviluppano ed usano software libero, hanno portato alla creazione di fondamentali beni comuni digitali.

Negli ultimi anni si sono costruiti beni comuni digitali costituiti da opere diverse dal software, come per esempio *Wikipedia* e *Open Street Map*.

⁴³ Vedi <http://www.ohwr.org/projects/cernohl/wiki>

⁴⁴ Vedi <http://www.tapr.org/ohl.html>

⁴⁵ L'art. 25 (Protezione dei dati fin dalla progettazione e protezione per impostazione predefinita) del Regolamento, tra l'altro, stabilisce che "...sia al momento di determinare i mezzi del trattamento sia all'atto del trattamento stesso il titolare del trattamento mette in atto misure tecniche e organizzative adeguate, quali la pseudonimizzazione, volte ad attuare in modo efficace i principi di protezione dei dati, quali la minimizzazione, e a integrare nel trattamento le necessarie garanzie al fine di soddisfare i requisiti del presente regolamento e tutelare i diritti degli interessati.". Il considerando 78 dello stesso Regolamento precisa che "...In fase di sviluppo, progettazione, selezione e utilizzo di applicazioni, servizi e prodotti basati sul trattamento di dati personali o che trattano dati personali per svolgere le loro funzioni, i produttori dei prodotti, dei servizi e delle applicazioni dovrebbero essere incoraggiati a tenere conto del diritto alla protezione dei dati allorché sviluppano e progettano tali prodotti, servizi e applicazioni e, tenuto debito conto dello stato dell'arte, a far sì che i titolari del trattamento e i responsabili del trattamento possano adempiere ai loro obblighi di protezione dei dati. I principi della protezione dei dati fin dalla progettazione e di default dovrebbero essere presi in considerazione anche nell'ambito degli appalti pubblici."

L'interazione tra licenze libere e pratiche sociali coerenti con il sistema valoriale delle comunità del software libero, con gli adattamenti che, pragmaticamente, possano essere individuati come utili, potrebbe portare allo sviluppo di nuovi beni comuni digitali, costituiti da contenuti digitali, banche di dati e/o disegni digitali di oggetti materiali.

Bibliografia

- Bain, Malcom. "Software Interactions and the GNU General Public License" *International Free and Open Source Software Law Review*, 2-2 (2010). Accessed June 15, 2017. Available at <http://www.ifosslr.org/ifosslr/article/view/44>
- Elliott, Mark. "Stigmergic Collaboration: The Evolution of Group Work" *M/C Journal*, 9-2 (2006). Accessed May 16, 2017. Available at <http://journal.media-culture.org.au/0605/03-elliott.php>
- Hemel, Armijn, and Coughlan, Shane. *Practical GPL Compliance*. San Francisco, CA: Linux Foundation, 2017.
- Kuhn, Bradley M., and Sebros, Anthony K. Jr., and Gingerich, Denver, and Free Software Foundation, Inc., and Software Freedom Law Center. *Copyleft and the GNU General Public License: A Comprehensive Tutorial and Guide*, Accessed May 16, 2017. Available at <https://copyleft.org/guide/>
- Fontana, Richard, and Kuhn, Bradley M., and Moglen, Eben, and Norwood, Matthew, and Ravicher, Daniel B., and Sandler, Karen, and Vasile, James, and Williamson, Aaron. *A Legal Issues Primer for Open Source and Free Software Projects*, Accessed May 16, 2017. Available at <http://softwarefreedom.org/resources/2008/foss-primer.pdf>
- Margoni, Thomas. "Not for Designers: On the Inadequacies of EU Design Law and How to Fix It" *Journal of Intellectual Property, Information Technology and E-Commerce Law* 4.3 (2013): 225-248. Accessed May 16, 2017. Available at <http://www.jipitec.eu/issues/jipitec-4-3-2013/3845/margoni.pdf>
- Weber, Steven. *The Success of Open Source*. Cambridge, Massachusetts: Harvard University, 2004.
- Meeker, Heather. *Open source for business. A practical guide to open source licensing*. North Charleston SC: Createspace Independent Publishing Platform, 2017.
- Metzger, Axel. *Free and Open Source Software (FOSS) and other Alternative License Models: A Comparative Analysis*. Switzerland: Springer International, 2016.
- Rosen, Lawrence. *Open source licensing: software freedom and intellectual property law*. Upper Saddle River, NJ: Prentice Hall Professional Technical Reference, 2005.
- Van den Brande, Ywein, and Xia, Yang, and von Willebrand, Martin, and Perbost, Fabrice, and Walter, Alan, Engelhardt, Tim, and Jaeger, Till, and Greenbaum, Eli, and Piana, Carlo, and Aliprandi, Simone, and Dammers, Wouter, and van Kerkvoorden, Wanda, and Rychlicki, Tomasz, and Bain, Malcolm, and Jaccard, Michel, and Ancelle, Juliette, and Katz, Andrew, and Webbink, Mark H. *The International Free and Open Source Software Law Book*. Open Source Press GmbH, 2011. Accessed May 16, 2017. Available at <http://ifosslawbook.org>