

IX CONVEGNO ANNUALE DELL' ASSOCIAZIONE ITALIANA DEI PROFESSORI
UNIVERSITARI DI DIRITTO COMMERCIALE "ORIZZONTI DI DIRITTO
COMMERCIALE"

"PROBLEMI ATTUALI DELLA PROPRIETÀ NEL DIRITTO COMMERCIALE"
ROMA, 23-24 febbraio 2018

FILIPPO CALDA BECCADELLI

**I diritti di esclusiva sul software e sulle informazioni ottenute a seguito
di reverse engineering alla luce della nuova Direttiva Trade Secret**

SOMMARIO: 1. La nozione di software e l'attività di reverse engineering 2. L'oggetto della tutela
autorale nel software 3. Il contenuto dell'esclusiva sul software ed i limiti ai poteri del titolare 4.
La nuova Direttiva europea sulle informazioni segrete. 5. Osservazioni sul coordinamento degli
artt. 5 e 6 della Direttiva Software con la Direttiva sulle informazioni segrete 6. Conclusioni.

1. La nozione di software e l'attività di reverse engineering

A differenza del c.d. hardware, assimilabile ad un qualsiasi prodotto industriale di natura meccanica (nozione che comprende, tra gli altri, computer, cellulari e tablet), il programma per elaboratore, conosciuto anche come software (nel prosieguo verranno utilizzati i due termini indistintamente anche se dal punto di vista tecnico corrispondono a due concetti molto diversi fra loro¹) presenta invece una natura giuridica "ibrida", in quanto è formato sia da una sequenza logica di istruzioni, contenenti uno o più algoritmi, in forma di procedura immateriale, sia dai mezzi materiali sui quali queste istruzioni vengono tradotte ed incorporate, ossia dei supporti hardware². La definizione più appropriata di software è certamente quella contenuta nel *Copyright Act* americano, che definisce software come "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result"³.

Il programma è normalmente scritto dall'uomo nel c.d. codice sorgente per poi essere tradotto dall'elaboratore in una serie di bit (il bit è quell'unità di misura dell'informatica espressa in forma binaria: sequenze

¹ Con il termine software si intende l'insieme composto dal "sistema operativo" e da tutti i programmi che su di esso si appoggiano, ovvero l'aspetto dinamico del processore che costituisce un'entità più complessa rispetto al programma che ne è una componente. Si vedano Per un glossario informatico si veda L. BOSOTTI, *Glossario*, L.C. UBERTAZZI (a cura di), *La legge sul software*, Giuffrè, Milano, 1994, p. 310.

² M.A. CARUSO, *Disciplina giuridica del software e interesse della collettività*, Giuffrè, Milano, 1989, p. 9. Oggi accade peraltro sempre più spesso che un programma non sia nemmeno contenuto su un supporto fisico e venga acquistato direttamente on-line per poi essere scaricato sul computer dell'utente.

³ Titolo 17, Capitolo 1, Paragrafo 101, United States Code.

di "0" e "1"), ovvero nel c.d. codice oggetto⁴: il codice sorgente è scritto su un supporto materiale (ad esempio su un disco) in linguaggio di programmazione ad alto livello di astrazione comprensibile all'uomo ma non alla macchina. Per essere leggibile da parte dell'elaboratore, il codice sorgente deve essere tradotto dal compilatore nel c.d. codice oggetto, un linguaggio leggibile da parte della macchina ma non da parte dell'uomo⁵.

La fase appena descritta è la c.d. fase di compilazione, ossia l'operazione che converte le istruzioni di un programma scritto in un linguaggio di programmazione (sorgente) in un codice oggetto (detto anche eseguibile) suscettibile di essere caricato ed eseguito da un elaboratore e dal relativo Sistema Operativo⁶.

Da quest'ultima forma del programma è molto improbabile poter risalire al codice sorgente⁷, o meglio, è possibile attraverso la c.d. "decompilazione" - un'operazione di *reverse engineering* diretta a ricostruire il codice sorgente partendo dal codice oggetto (si tratta dell'operazione opposta alla citata compilazione) - risalire ad una versione approssimata del codice originario (c.d. pseudo-codice)⁸. Nonostante l'incompletezza del risultato che si ottiene dalla decompilazione, lo pseudo-codice sorgente che si ottiene da questa operazione può essere idoneo a fornire molti dati utili relativi al programma decompilato⁹.

È bene evidenziare come l'attività di decompilazione rientri nella nozione più ampia di *reverse engineering* che si esplica in quell'attività consistente nello "*...starting with the known product and working backward to divine the process which aided in its development or manufacture*"¹⁰.

Le società operanti nel settore dello sviluppo del software svolgono attività di *reverse engineering* quando non sono a conoscenza del codice sorgente di un programma e debbono risalire al medesimo. Ciò può, ad esempio, accadere quando (i) si voglia migliorare il programma e sviluppare una nuova versione del medesimo ma questo sia stato acquistato da un terzo che non ha fornito il codice sorgente o sia stato sviluppato da programmatori che non sono più nella società e non hanno lasciato alcuna traccia scritta del loro lavoro (ii) la necessità di correggere

⁴ R. BORRUSO, *La tutela giuridica del software*, Giuffrè, Milano, 1999, p. 7. Si veda anche G. GUGLIELMETTI, *L'invenzione di software*, Giuffrè, Milano, 1997, p. 198..

⁵ CALÀ L., *Software come opera letteraria all'inizio degli anni novanta*, MUSSO A. (a cura di), *Il software tra brevetto e diritto d'autore. Regime "proprietario" o open source?*, Bononia University Press, Bologna, 2005, pp. 56-71.

⁶ L. BOSOTTI, cit. (nt.1), p. 305.

⁷ L. BORRUSO, cit. (nt. 5), p.8.

⁸ L. BOSOTTI, cit. (nt. 1), p. 306.

⁹ G. GUGLIELMETTI, *Analisi e decompilazione dei programmi*, L.C. UBERTAZZI (a cura di), *La legge sul software*, Giuffrè, Milano, 1994, p. 154.

¹⁰ La definizione è tratta dal noto caso americano *Kewanee Oil Co. v. Bicron Corp.*, 416 U.S. 470, 476, 1974.

degli errori del programma o di adattarlo a proprie specifiche esigenze interne (iii) la necessità di ottenere le interfacce necessarie per sviluppare un prodotto che possa comunicare ed essere compatibile (c.d. interoperabilità) con il programma studiato (iv) la necessità di ottenere le interfacce necessarie per sviluppare un programma concorrente (v) la necessità di ottenere informazioni per ragioni completamente diverse da quella dell'ottenimento dell'interoperabilità, quali a titolo esemplificativo, la comprensione delle caratteristiche e delle performance di un determinato programma¹¹.

Da quanto sopra emerge che una delle ragioni per cui si ricorre al *reverse engineering* di un software è quello della ricerca delle c.d. interfacce di un programma ai fini di consentirne l'interoperabilità con un altro¹².

L'esempio che spesso viene portato dalla dottrina per spiegare il fenomeno della c.d. interoperabilità è quello della compatibilità fra una spina e una presa elettrica, per cui la forma della spina deve essere esattamente quella compatibile con la presa, altrimenti il dispositivo ad essa collegato non funzionerà¹³. La forma della spina e la presa possono essere considerate le interfacce in assenza delle quali i programmi non potranno comunicare tra di loro (ovvero raggiungere la c.d. interoperabilità).

Un altro importante fenomeno che costituisce un limite alla possibilità di decompilare un programma è quello del c.d. *software as a service* (SaaS): sempre più programmi sono on-line e/o sono forniti semplicemente come servizi ("*as a service*") per cui l'utente entra in contatto con la sola interfaccia grafica e non con il codice oggetto che si trova nel server a cui ovviamente l'utente non ha accesso.

In certi casi (il SaaS ne è un esempio) è quindi possibile che il codice sorgente non sia a disposizione dell'utilizzatore: in questo caso la "tecnica" di *reverse engineering* che normalmente viene adottata è la c.d. *black box analysis*, operazione consistente nell'osservare il funzionamento del programma, attraverso, ad esempio, l'inserimento di input al fine di comprendere i corrispondenti output del medesimo¹⁴.

A prescindere dai limiti entro i quali può essere effettuato, che si andranno ad esaminare nel prosieguo, il *reverse engineering* costituisce il più delle volte un'attività che porta ad innovare piuttosto che a clonare.

¹¹ J. BAND - M. KATOH, *Interface on Trial - Intellectual Property and Interoperability in the Global Software Industry*, Boulder, Westview Press, Inc., 1995, p. 17.

¹² A. JOHNSON-LAIRD, *Reverse Engineering of Software Separating Legal Mythology from Actual Technology*, in *Software Law Journal*, aprile 1992, pp. 350-353.

¹³ P. SAMUELSON, *The Strange Odyssey of Software Interfaces and Intellectual Property Law*, in <http://escholarship.org/uc/item/59n441jv>, 2008, p. 1; J. BAND - M. KATOH, cit. (nt. 12), p. 7.

¹⁴ A. VON ROOIJEN, *The software interface between copyright and competition law*, Wolters Kluwer, Alphen aan den Rijn, 2010, pp. 18-19.

Infatti, se il fine ultimo fosse quello di clonare, vi sarebbe la più semplice opzione consistente nella copiatura diretta del codice oggetto (questa condotta costituisce di regola una violazione del diritto d'autore): oggi l'operazione è semplicissima e permette di copiare il programma senza necessità di conoscere il codice sorgente, portando ad un risultato perfetto dato che la copia corrisponde in ogni bit all'originale. Questa "soluzione" è normalmente adottata da chi non ha il problema di dover comprendere e/o controllare e/o modificare e/o implementare il programma¹⁵.

2. L'oggetto della tutela autoriale nel software

Prima di esaminare la disciplina EU relativa alla tutela del software è opportuno delineare il quadro della tutela autoriale del software a livello di convenzioni internazionali. In particolare, rilevano la Convenzione per la protezione delle opere letterarie ed artistiche (di seguito Convenzione di Berna) e i TRIPs *Argement* (di seguito TRIPs).

Ai sensi dell'articolo 2, comma 1, della Convenzione di Berna (nella versione del 28 settembre 1979) "*L'espressione "opere letterarie ed artistiche" comprende tutte le produzioni nel campo letterario (...), qualunque ne sia il modo o la forma di espressione (...)*".

L'art. 10, comma 1, TRIPs prevede poi una norma dedicata alla protezione del software attraverso le norme della Convenzione di Berna: "*I programmi per elaboratore, in codice sorgente o in codice oggetto, sono protetti come opere letterarie ai sensi della Convenzione di Berna*".

Il Trattato dell'Organizzazione mondiale per la proprietà intellettuale (OMPI/WIPO) sul diritto d'autore, adottato a Ginevra il 20 dicembre 1996 ed entrato in vigore, per quanto riguarda l'Unione europea, il 14 marzo 2010, ha una norma analoga all'art. 10 TRIPs, con l'unica differenza che non vi è un espresso riferimento alla tutela del codice sorgente e del codice oggetto come avviene invece nei TRIPs¹⁶.

Sulla base del suddetto quadro internazionale, si comprende dunque la base giuridica dell'inquadramento del programma per elaboratore come opera letteraria: si tratta di una vera e propria *factio iuris*, visto che il codice sorgente, ma soprattutto il codice oggetto, sono qualcosa di molto diverso rispetto ad un'opera letteraria. Infatti, mentre un'opera letteraria comunica qualcosa al lettore, il codice oggetto è incomprendibile ad un lettore "medio". Allo stesso modo il codice sorgente è comprensibile solo ad un programmatore a conoscenza di quel determinato linguaggio e quindi certamente ad un numero maggiore di lettori rispetto al codice oggetto ma non certamente ad un lettore "medio".

¹⁵ R. BORRUSO, cit. (nt. 5), p. 8-9.

¹⁶ L'inquadramento internazionale è ripreso dalla nota decisione della Corte di Giustizia C-406/2010, *SAS Institute Inc, Vs. World Programming Ltd.,* 2 maggio 2012.

La disciplina di riferimento a livello EU per quanto riguarda la tutela autoriale del software è contenuta nella Direttiva europea 91/250 (poi sostituita dalla Direttiva 2009/24: di seguito Direttiva Software) alla quale si sono conformate le rispettive legislazioni nazionali. In Italia la Direttiva Software è stata recepita con il d.lgs. n. 518/1992 che ha introdotto, fra gli altri, gli artt. 2, n. 8, 64-bis, ter e quater della legge sul diritto d'autore (l.d.a.).

A differenza della normativa americana, di cui è stata riportata l'interessante definizione di software, la Direttiva europea non ne fornisce alcuna definizione, limitandosi a rimandare alla Convenzione di Berna, che, come visto, inquadra il software nella categoria di opere letterarie.

Nonostante non vi sia una definizione che identifichi chiaramente l'oggetto della tutela autoriale, è tuttavia pacifica, sia in dottrina sia in giurisprudenza, la tutelabilità con il diritto d'autore non solo del codice sorgente ma anche del codice oggetto¹⁷.

Di particolare interesse, ai fini dell'identificazione dell'ambito di tutela del software è il fatto che la Direttiva Software include nella nozione (come visto non definita) di programma per elaboratore anche "*il materiale preparatorio per la progettazione di un programma*" (art. 1, comma 1, Direttiva Software), a condizione "*che siano di natura tale da consentire la realizzazione di un programma per elaboratore in una fase successiva*" (Considerando 7, Direttiva Software)¹⁸.

In dottrina si è pertanto ipotizzata la violazione del diritto d'autore sui lavori preparatori nel caso in cui un concorrente utilizzi tale materiale per scrivere un programma¹⁹, anche se ciò pare un caso più di scuola, visto che nel mondo reale risulterà molto difficile accertare la violazione del diritto d'autore, essendo invece più facilmente riconducibile alla fattispecie di sottrazione di informazioni riservate.

¹⁷ Per l'espressione di tale principio nell'ambito della giurisprudenza europea si rimanda alla decisione della Corte di Giustizia nel caso C-393/2009, *Bezpečnostní softwarová asociace – Svaz softwarové ochrany Vs. Ministerstvo kultury*, 22 dicembre 2010.

¹⁸ In particolare, secondo tale considerando "*Ai sensi della presente direttiva, il termine «programma per elaboratore» indica programmi in qualsiasi forma, compresi quelli incorporati nell'hardware; questo termine comprende anche i lavori preparatori di progettazione per realizzare un programma, a condizione che siano di natura tale da consentire la realizzazione di un programma per elaboratore in una fase successiva*". Il "*materiale preparatorio*" è stato definito dalla giurisprudenza di merito nostrana come quel materiale "*che deve presentare un certo livello di dettaglio, così da consentire ad un programmatore esperto di passare alla stesura del sorgente del programma senza compiere alcun ulteriore sforzo progettuale*" (Tribunale Roma, 6 maggio 2009, n. 9696, decisione tratta dalla banca dati De Jure).

¹⁹ P. SAMUELSON- T. VINJE – W. CORNISH, *Does copyright protection under the EU Software Directive extend to computer program behavior, languages and interfaces?*, in *European Intellectual Property Review*, marzo 2012, p. 159.

Come si vedrà nel prosieguo, questo è un punto da tenere ben presente, in quanto potenzialmente in grado di spostare il confine dell'estensione della tutela autoriale del software.

Passando ai requisiti per la tutelabilità di un software, la Direttiva Software prevede la tutela autoriale per i soli programmi dotati di originalità, ovvero se il programma "è il risultato della creazione intellettuale dell'autore" (art. 1, comma 3, Direttiva Software)²⁰.

Affinché un programma sia dotato di originalità, non è anzitutto necessario che il lavoro sia unico o di alta qualità²¹. Di regola, in ambito software, l'originalità di un programma non viene valutata sulla base di "meriti qualitativi o estetici del programma" (Considerando 8, Direttiva Software), bensì sulla base del numero di opzioni che un programmatore ha per scrivere un programma: due programmatori implementeranno infatti le medesime funzionalità utilizzando due espressioni diverse. Lo spazio di manovra che il programmatore ha, costituisce quindi il parametro per valutare l'originalità di un programma²².

Ai fini della protezione con il diritto d'autore non è pertanto necessario che l'opera sia nuova o abbia un certo livello "estetico" particolare, essendo sufficiente che si tratti di un'opera non copiata²³, o per usare un termine caro alla dottrina e alla giurisprudenza italiana, è necessario che l'opera porti con sé l'"impronta" del suo autore²⁴.

Al fine dell'individuazione dell'oggetto tutelabile, è altresì rilevante ricordare il principio generale sulla base del quale il diritto d'autore protegge l'espressione e non l'idea. Tale principio è specificato dall'art. 1, comma 2, Direttiva Software, secondo cui "Le idee e i principi alla base di qualsiasi elemento di un programma per elaboratore, compresi quelli alla base delle sue interfacce, non sono tutelati dal diritto d'autore"²⁵. Salta subito all'occhio come la normativa europea faccia un espresso riferimento anche

²⁰ Art. 1, comma 3, Direttiva Software.

²¹ R. NIMMER, *Law of Computer Technology*, Thomson Reuters, St. Paul, 2015, Vol I, p. 1-30.

²² A. VON ROOIJEN, cit. (nt. 14), p. 58.

²³ R.C. DREYFUSS - R.R. KWALL, *Intellectual Property - Cases and Materials on Trademark, Copyright and Patent Law*, Foundation Press, New York, 2004, p. 223.

²⁴ Si veda ad esempio Corte di Appello Milano, 18 ottobre 1974 decisione citata in S. ERCOLANI, *L'oggetto della protezione: l'opera dell'ingegno*, C. GALLI - A.M. GAMBINO (a cura di), *Codice commentato della proprietà industriale e intellettuale*, Utet Giuridica, Torino, 2011, p. 2840.

²⁵ Art. 1, comma 2, Direttiva Software. In particolare nei Considerando della Direttiva si legge al Considerando 11 che, conformemente al principio secondo cui solo l'espressione di un programma per elaboratore è tutelata dal diritto d'autore, le idee e i principi che sono alla base della logica, degli algoritmi e dei linguaggi di programmazione non sono tutelati a norma della direttiva in questione. Il Considerando aggiunge poi che, conformemente alla legislazione e alla giurisprudenza degli Stati membri, nonché alle convenzioni internazionali sul diritto d'autore, l'espressione di tali idee e principi deve essere tutelata dal diritto d'autore.

alle interfacce. Tale specificazione è il risultato di un lungo percorso e degli intensi dibattiti che hanno portato all'adozione della Direttiva.

Un importante corollario derivante dal suddetto principio è quello espresso dalla c.d. *Merger Doctrine*, secondo cui se l'idea e l'espressione non sono scindibili, o comunque nel caso in cui l'idea possa essere espressa solo in un numero limitato di modalità, allora l'idea non sarà proteggibile. Tale principio è stato espresso in ambito europeo dalla Corte di Giustizia nel caso *Bezpečnostní Vs. Ministerstvo kultury*, in cui al punto 49 della decisione si legge che *"...il criterio dell'originalità non sarebbe soddisfatto, in quanto i vari modi di realizzare un'idea sono così limitati che l'idea e l'espressione si confondono"*²⁶.

3. Il contenuto dell'esclusiva sul software ed i limiti ai poteri del titolare

Come espresso dall'art. 4 della Direttiva Software (e dal corrispondente art. 64-bis l.d.a.), il titolare del software ha il diritto di effettuare o autorizzare le seguenti operazioni²⁷:

- *"la riproduzione, permanente o temporanea, totale o parziale di un programma per elaboratore con qualsivoglia mezzo, in qualsivoglia forma. Nella misura in cui operazioni come il caricamento, la visualizzazione, l'esecuzione, la trasmissione o la memorizzazione del programma per elaboratore richiedono una riproduzione, tali operazioni devono essere sottoposte ad autorizzazione da parte del titolare del diritto"*(art. 4, comma 1, lett. a));
- *"la traduzione, l'adattamento, l'adeguamento e ogni altra modifica di un programma per elaboratore e la riproduzione del programma che ne risulti, fatti salvi i diritti della persona che modifica il programma"* (art. 4, comma 1, lett. b));
- *"qualsiasi forma di distribuzione al pubblico, compresa la locazione, del programma per elaboratore originale e di copie dello stesso"* (art. 4, comma 1, lett. c)).

I poteri conferiti al titolare del software dall'art. 4, trovano tuttavia dei limiti negli artt. 5 e 6 della Direttiva Software (e nei corrispondenti artt. 64-ter e 64-quater l.d.a.).

In particolare, l'art. 5 della Direttiva Software ("Deroghe alle attività riservate") dispone ai suoi commi 1 e 3 che *"1. Salvo disposizioni contrattuali specifiche, non sono soggetti all'autorizzazione del titolare del diritto gli atti indicati nell'articolo 4, paragrafo 1, lettere a) e b), allorché tali atti sono necessari per un uso del programma per elaboratore conforme alla sua destinazione, da parte del legittimo acquirente, nonché per la correzione di errori.*

...

²⁶ C-393/2009, *Bezpečnostní softwarová asociace – Svaz softwarové ochrany Vs. Ministerstvo kultury*, 22 dicembre 2010.

²⁷ Art. 4 Direttiva Software.

3. *La persona che ha il diritto di utilizzare una copia di un programma può, senza chiederne l'autorizzazione al titolare del diritto, osservare, studiare o sperimentare il funzionamento del programma, allo scopo di determinare le idee e i principi su cui è basato ogni elemento del programma, quando essa effettua le operazioni di caricamento, visualizzazione, esecuzione, trasmissione o memorizzazione del programma che ha il diritto di effettuare*²⁸.

Il primo comma dell'art. 5 della Direttiva Software prevede anzitutto la possibilità per il legittimo acquirente del programma, di effettuare un'attività di *reverse engineering* sul programma per la correzione di errori contenuti nel medesimo: la norma si riferisce infatti alla possibilità di effettuare tutte le operazioni di cui alla lett. a) e b) dell'art. 4 "*per un uso del programma ... conforme alla sua destinazione ..., nonché per la correzione di errori*": la norma prevede evidentemente la possibilità di decompilare (la sola *black box analysis* non sarebbe sufficiente) il codice del programma per poter effettuare un'attività di manutenzione consistente nella correzioni degli errori (c.d. manutenzione correttiva)²⁹.

In merito alla possibilità di limitare contrattualmente le attività sopra menzionate, come statuito dalla medesima norma - "*Salvo disposizioni contrattuali specifiche*" -, è stato rilevato come ciò appaia in netto contrasto con il Considerando 13 della Direttiva, secondo cui non possono essere vietati contrattualmente "*... gli atti di caricamento e di svolgimento necessari per l'utilizzazione di una copia di un programma legittimamente acquisita e l'atto di correzione dei suoi errori*"³⁰: dovrà dunque prevalere l'interpretazione più ragionevole della norma, anche alla luce del Considerando esaminato, visto che in caso contrario vi sarebbero limiti troppo eccessivi alla fruibilità del programma da parte del legittimo acquirente³¹. Pertanto, si dovrebbe leggere l'art. 8, che prevede la nullità di qualsiasi disposizione contrattuale "*non conforme all'articolo 6 o alle eccezioni di cui all'articolo 5, paragrafi 2 e 3*", come estendibile anche al primo comma dell'art. 5.

²⁸ Art. 5, commi 1 e 3, Direttiva Software.

²⁹ A. MUSSO, *Elaborazioni e programmi derivati*, L.C. UBERTAZZI (a cura di), *La legge sul software - commentario sistematico*, Giuffrè, Milano, 1994, p. 77; E. AREZZO, *Programmi per elaboratore*, C. GALLI - A.M. GAMBINO, *Codice commentato della proprietà industriale e intellettuale*, Utet Giuridica, Torino, 2011, p. 3103.

³⁰ Considerando 13 Direttiva Software.

³¹ T. VINJE, *Compliance with Article 85 in Software Licensing*, in *Law, Computer & Artificial Intelligence*, vol. 1, n. 2, 1992, pp. 172-174. Anche la dottrina italiana più autorevole si è espressa nel senso dell'inammissibilità delle limitazioni contrattuali alle attività di cui all'art. 5 della Direttiva: G. GUGLIELMETTI, *L'invenzione di software*, cit. (nt. 5), p. 209; R. BORRUSO, cit. (nt. 5), p. 55 e ss e 64; E. AREZZO, cit. (nt. 31), p. 3103.

Per le suddette ragioni l'incipit della norma in esame - "Salvo disposizioni contrattuali ..." - va interpretata in modo restrittivo, nel senso che non sono ammissibili restrizioni contrattuali alle attività di *reverse engineering* necessarie (che potranno includere anche attività di decompilazione) per la manutenzione del software.

L'altra norma fondamentale in relazione alla possibilità di effettuare attività di *reverse engineering* è il comma 3 dell'art. 5, che nella proposta originaria costituiva l'unica attività di *reverse engineering* consentita. Secondo il comma 3 dell'art. 5, che disciplina la c.d. *black box analysis*, colui che ha il diritto di utilizzare una copia di un programma può effettuare tale procedura senza necessità di ottenere alcuna autorizzazione da parte del titolare.

In particolare, l'attività che può essere svolta senza previa autorizzazione del titolare è quella di "osservare, studiare o sperimentare il funzionamento del programma allo scopo di determinare le idee e i principi su cui è basato ogni elemento del programma, quando essa effettua le operazioni di caricamento, visualizzazione, esecuzione, trasmissione o memorizzazione del programma che ha il diritto di effettuare" (art. 5, comma 3, Direttiva Software).

Le idee ed i principi di cui alla sopra citata norma, includono evidentemente anche le interfacce necessarie per ottenere l'interoperabilità con altri programmi.

Come rilevato da autorevole dottrina nostrana, la norma finisce di fatto per regolamentare una situazione che in astratto non lo dovrebbe essere visto che la *black box analysis* permetterà di risalire a idee e principi che di per sé dovrebbero essere di pubblico dominio e non tutelabili attraverso il diritto d'autore³².

Una questione non affrontata dalla norma in esame - diversamente dall'art. 6 della Direttiva Software, che disciplina specificamente l'uso delle informazioni ottenute dalla decompilazione - è quella dell'uso che può essere fatto delle informazioni ottenute dall'attività di *black box analysis* - la stessa domanda può essere posta in relazione alle informazioni ottenute a seguito dell'attività di manutenzione di cui al comma 1 dell'art. 5 - , si può dunque ritenere che nella Direttiva Software non vi sia alcuna particolare restrizione all'uso delle informazioni così ottenute³³.

³² E. AREZZO, cit. (nt. 31), p. 3103.

³³ La problematica viene sollevata da G. GUGLIELMETTI, cit. (nt. 10), p. 161. L'Autore risponde al quesito rilevando come "La lettera della norma non sembra contemplare anche questo genere di patti; essi tuttavia, implicano una restrizione contrattuale della libertà dell'utente di utilizzare le informazioni tecniche lecitamente acquisite, che potranno essere oggetto di valutazione in base alla disciplina antitrust". Si pone il problema anche T. VINJE, *Interoperable*

L'art. 5 non è certamente il massimo della chiarezza, tuttavia, come si vedrà meglio nel prosieguo, la domanda su come tutelare le informazioni così ottenute può probabilmente trovare una risposta nella nuova Direttiva europea sulle informazioni segrete.

L'altra norma che pone dei limiti ai diritti esclusivi del titolare del diritto è il già citato art. 6 della Direttiva Software, che ammette la decompilazione ai fini dell'ottenimento dell'interoperabilità fra programmi.

In particolare, l'art. 6 della Direttiva Software è diviso in due commi: mentre il primo comma detta le condizioni che devono essere rispettate per poter decompilare legittimamente un programma, il secondo comma indica i limiti entro i quali possono essere utilizzate le informazioni ricavate dalla decompilazione. Si riporta qui di seguito il contenuto dei primi due commi dell'art. 6: *"1. Per gli atti di riproduzione del codice e di traduzione della sua forma ai sensi dell'articolo 4, paragrafo 1, lettere a) e b), non è necessaria l'autorizzazione del titolare dei diritti qualora l'esecuzione di tali atti al fine di modificare la forma del codice sia indispensabile per ottenere le informazioni necessarie per conseguire l'interoperabilità con altri programmi di un programma per elaboratore creato autonomamente, purché sussistano le seguenti condizioni:*

a) tali atti siano eseguiti dal licenziatario o da un'altra persona che abbia il diritto di utilizzare una copia del programma o, per loro conto, da una persona abilitata a tal fine;

b) le informazioni necessarie per ottenere l'interoperabilità non siano già facilmente e rapidamente accessibili alle persone indicate alla lettera a); e

c) gli atti in questione siano limitati alle parti del programma originale necessarie per conseguire l'interoperabilità.

2. Le disposizioni del paragrafo 1 non consentono che le informazioni ottenute in virtù della sua applicazione:

a) siano utilizzate a fini diversi dalla realizzazione dell'interoperabilità del programma creato autonomamente;

b) siano comunicate a terzi, fatta salva la necessità di consentire l'interoperabilità del programma creato autonomamente;

c) siano utilizzate per lo sviluppo, la produzione o la commercializzazione di un programma sostanzialmente simile nella sua espressione, o per ogni altro atto che violi il diritto di autore.

...".

La violazione del primo comma della norma in esame – anche di una sola delle tre condizioni contenute nel medesimo – prefigura una vera e propria violazione del diritto d'autore, mentre il mancato rispetto delle limitazioni relative all'uso delle informazioni ottenute sulla base del

Product development under the EC Software Directive, in Computer Law & Practice, vol. 8, n. 6, 1992, p. 194.

comma 1 dell'art. 6 della Direttiva, regolata dal successivo comma 2, non costituisce alcuna violazione del diritto d'autore e non fa venire meno la legittimità dell'operazione di decompilazione effettuata sulla base del comma 1 della medesima norma, salvo quanto disposto dalla lett. c) che prevede un'ipotesi di violazione del diritto d'autore.

Non essendo presente nella Direttiva Software alcuna definizione di decompilazione, si farà riferimento alla decompilazione come a quell'attività di *reverse engineering* che, come già definita in precedenza consente di risalire ad uno pseudo codice sorgente partendo dal codice oggetto (si tratta dell'operazione opposta alla compilazione).

Nella Direttiva è invece individuabile al Considerando 10 una definizione di interoperabilità: l'interoperabilità è la "... capacità di due o più sistemi di scambiare informazioni e di usare reciprocamente le informazioni scambiate". Il Considerando 10, nello spiegare la "funzione di comunicare e operare con altri componenti di un sistema informatico e con gli utenti" propria dei programmi per elaboratore, si riferisce altresì alla necessità di "consentire a tutti i componenti software e hardware di operare con altri software e hardware e con gli utenti in tutti i modi in cui sono destinati a funzionare".

Per quanto attiene allo stato di avanzamento del programma oggetto di decompilazione, anche se la Direttiva include nella nozione di software il materiale preparatorio (art. 1, comma 1), è ragionevole ritenere che si tratti di un programma già funzionante e nella versione finale. In caso contrario, vista l'estrema incertezza sulle caratteristiche che avrà il programma finale, evidentemente il *reverse engineer* non si arrischierà ad intraprendere un'attività di decompilazione, anche alla luce degli elevati costi che essa comporta.

L'interoperabilità tra software a cui si riferisce l'art. 6 della Direttiva Software non è solo quella fra il programma originale e il programma che si sta sviluppando ai fini di trovare una compatibilità con il primo, bensì anche quella con programmi diversi da quello decompilato: ad esempio, come nel caso in cui il secondo programma costituisca un sistema operativo realizzato per competere direttamente con l'originale e debba quindi essere compatibile anche con gli altri programmi operanti sull'originale³⁴.

L'art. 6 non ammette poi la decompilazione ai fini della manutenzione del software, essendo limitato al solo ottenimento dell'interoperabilità: il timore, spiegato anche nel Memorandum esplicativo che accompagna la proposta modificata della Commissione, era quello di espandere eccessivamente il potere di decompilare, ciò anche

³⁴ W. BLOCHER - M. WALTER, *Computer Program Directive*, M. WALTER - S. VON LEWINSKI (a cura di), *European Copyright Law - A Commentary*, Oxford University Press, New York, 2010, pp. 172-173.

alla luce dell'ampiezza del termine "manutenzione". L'attività di *reverse engineering* ai fini di manutenzione è comunque, come si è già visto, ammissibile, quanto meno, per la manutenzione correttiva, alla luce dell'art. 5, comma 1, Direttiva Software³⁵.

È interessante rilevare come il secondo comma dell'art. 6 della Direttiva preveda, a differenza di quanto statuito per le informazioni ottenute attraverso la *black box analysis* (art. 5, comma 3) o comunque a seguito di manutenzione del programma (art. 5, comma 1), delle stringenti restrizioni relativamente all'uso delle informazioni ottenute a seguito dell'applicazione del primo comma dell'art. 6 della Direttiva Software.

Ovviamente tale norma è applicabile solo nel caso in cui le informazioni siano state ottenute nel rispetto del primo comma dell'art. 6 della Direttiva Software.

Dall'analisi della norma emerge tuttavia come non vi sia una precisa definizione di "informazioni". Tale definizione è desumibile dal Considerando 10 che identifica molto genericamente le informazioni di cui all'art. 6 con il "*materiale per consentire a tutti i componenti software e hardware di operare con altri software e hardware e con gli utenti in tutti i modi in cui sono destinati a funzionare*".

Si tratta di un concetto molto ampio di "informazione", non essendovi alcuna espressa esclusione dalla nozione di informazione né dei principi né delle idee. In tal modo verrebbe quindi fornita una tutela che va oltre la classica tutela autoriale e che si estenderebbe in teoria anche ai principi e alle idee che come espresso dall'art. 1, comma 2 della Direttiva non dovrebbero rientrare nella tutela autoriale³⁶.

In particolare, le restrizioni alla disponibilità delle informazioni lecitamente ottenute *ex art. 6, comma 1*, si riferiscono a tre possibili situazioni, disciplinate dalle lett. a), b) e c) del secondo comma dell'art. 6:

³⁵ J. BAND - M. KATOH, cit. (nt. 12), p. 249-250; M. SUCKER, *The Software Directive-Between the Combat against Piracy and the Preservation of Undistorted Competition*, M. LEHMANN - C. TAPPER (a cura di), *A Handbook of European Software Law*, Oxford University Press, New York, 1993, p. 18; G. GUGLIEMMETTI, cit. (nt. 10), pp. 180-181.

³⁶ T. VINJIE, cit. (nt. 35), p. 195; E. AREZZO, cit. (nt. 31), p. 3103; W. BLOCHER - M. WALTER, cit. (nt. 38), pp. 176-179. Questi ultimi due Autori (Blocher e Walter) hanno una visione più restrittiva del significato di informazioni ottenute a seguito della decompilazione: la posizione di Blocher e Walter nasce da una lettura sistematica e ritiene non incluse le informazioni che non rientrerebbero nello *scope of protection* del diritto d'autore. Pertanto a dire degli Autori la norma, che da una prima lettura sembra restrittiva, permette invece di utilizzare e diffondere liberamente (lo chiamano "free use") le informazioni che non rientrano nello *scope of protection* del diritto d'autore. La norma viene quindi così riletta (e reinterpretata): "*The information and code parts that have been obtained by decompilation according to paragraph 1 and are protected by copyright under this Directive are permitted to be ...*".

- la lett. a) non consente che le informazioni possano essere utilizzate per fini diversi dalla realizzazione dell'interoperabilità del programma creato autonomamente. In altre parole, il *reverse engineer* potrà utilizzare solo la parte delle informazioni necessarie per l'ottenimento dell'interoperabilità³⁷.
- La lett. b) non consente la comunicazione delle informazioni a terzi, salvo il caso in cui ciò sia necessario per consentire l'interoperabilità del programma creato autonomamente. Pertanto, sarà possibile la sola diffusione delle informazioni necessarie per permettere l'interoperabilità da parte di altri programmi con il programma creato autonomamente (in tal caso il *reverse engineer* potrà per esempio comunicare le informazioni necessarie per poter consentire l'interoperabilità con il suo programma)³⁸. Si deve rilevare come la comunicazione di tali informazioni potrebbe in astratto portare ad una conoscenza indiretta da parte del terzo di informazioni necessarie anche per ottenere l'interoperabilità con il primo programma oggetto della legittima decompilazione, risparmiando al medesimo tutta l'attività di *reverse engineering* effettuata.
- La lett. c) non consente che le informazioni siano "*utilizzate per lo sviluppo, la produzione o la commercializzazione di un programma sostanzialmente simile nella sua espressione o per ogni altro atto che violi il diritto d'autore*". La norma è chiara e non necessita di particolari spiegazioni; preme tuttavia rilevare come tale norma non escluda la possibilità di realizzare prodotti concorrenti che, pur divergendo dal punto di vista del codice di programmazione, potranno essere simili per quanto attiene agli aspetti funzionali³⁹.

Dall'esame delle lett. a), b) e c) emerge tuttavia una potenziale lacuna laddove il decompilatore realizzi un programma che non sia sostanzialmente simile ma che contenga parti di codice che non servono al fine di ottenere l'interoperabilità. In dottrina è stato rilevato come tale esclusione non sarebbe ragionevole in quanto porterebbe all'appropriazione di informazioni tecniche quali algoritmi che pur non rientrando nella forma espressiva tutelata dal diritto d'autore hanno un valore non solo economico, così facendo infatti "*il terzo ha abusivamente*

³⁷ J. BAND – M. KATOH, cit. (nt. 12), p. 253; W. BLOCHER – M. WALTER, cit. (nt. 38), p. 178. Secondo Blocher e Walter dalla lettura del comma 1 che si riferisce ad una interoperabilità con "altri programmi", deriva che le informazioni ottenute potranno essere legittimamente utilizzate per la ricerca di interoperabilità anche con altri programmi anche eventualmente diversi dal programmi originale.

³⁸ J. BAND – M. KATOH, cit. (nt. 12), p. 253.

³⁹ In senso contrario si veda A. JOHNSON-LAIRD, cit. (nt. 13), p. 331, secondo cui la nuova direttiva ha sostanzialmente messo fuori legge la possibilità di effettuare attività di *reverse engineering* per la realizzazione di prodotti concorrenti.

utilizzato informazioni segrete ricavate attraverso un'attività della legge considerata lecita soltanto per uno scopo diverso"⁴⁰.

L'art. 6, comma 2, finisce quindi con l'imporre una disciplina restrittiva all'uso delle informazioni così ottenute, assoggettandole ad un vero e proprio regime *sui generis*, visto che (i) va oltre alla tutela autoriale, tutelando anche elementi in astratto non proteggibili dal diritto d'autore (come algoritmi), e (ii) prevede dei limiti ulteriori rispetto alla tutela come informazione segreta rispetto alla possibilità di loro riutilizzo da parte del *reverse engineer*.

Il suddetto regime di tutela non dipende poi dalla presenza o meno dei requisiti che sono normalmente necessari per qualificare un'informazione come segreta, e alla cui sussistenza è legata anche la durata (potenzialmente illimitata) del diritto sulle informazioni segrete.

Tale regime *sui generis* è evidentemente anche conseguenza della prassi di non divulgare mai il codice sorgente e di consentirne quindi una tutela equiparabile a quella delle informazioni segrete, resa possibile anche dal fatto che non è presente un obbligo di divulgazione dell'opera⁴¹.

4. La nuova Direttiva europea sulle informazioni segrete

Come è stato giustamente osservato da autorevole dottrina italiana, *"il diritto d'autore sul software è costruito in modo da rafforzare la tutela del segreto. Il diritto d'autore infatti può operare come una sorta di «chiavistello giuridico», perché consente al titolare di vietare (al di fuori della specifica ipotesi della decompilazione per la ricerca delle interfacce ...) gli atti di riproduzione e di creazione di copie derivate che sono necessari ai terzi per studiare il programma in forma oggetto, concretamente immesso in commercio"*. In tal modo il diritto d'autore crea un ostacolo legale all'accesso alle informazioni segrete anche in relazione ad una attività di *reverse engineering* su un prodotto immesso sul mercato, normalmente ammessa in relazione all'apprendimento delle informazioni segrete in generale⁴².

Il 2016 è stato l'anno della tutela delle c.d. informazioni segrete o informazioni riservate (o *trade secrets*): sia negli USA, con il *Defendant Trade Secret Act* (di seguito DTSA), sia in Europa, con la Direttiva *"sulla protezione del know-how riservato e delle informazioni commerciali riservate (segreti commerciali) contro l'acquisizione, l'utilizzo e la divulgazione"* (Direttiva sulle informazioni segrete) sono infatti state emanate delle normative con il fine di "armonizzare" la normativa vigente in materia. Le

⁴⁰ G. GUGLIELMETTI, cit. (nt. 10), pp. 192-196.

⁴¹ E. AREZZO, cit. (nt. 31), p. 3103.

⁴² G. GUGLIELMETTI, *La proposta di Direttiva sulla brevettazione delle invenzioni in materia di software*, in *Riv. Dir. Ind.*, 2002, 5, pp. 462-463.

due normative hanno in comune il fatto di considerare legittime le attività di *reverse engineering* aventi ad oggetto informazioni segrete⁴³.

A livello di Convenzioni internazionali, la regolamentazione sulla tutela delle informazioni riservate è fornita anzitutto dagli Accordi TRIPs, ed in particolare dai commi 1 e 2 dell'art. 39 TRIPs, secondo cui: "1. *Nell'assicurare un'efficace protezione contro la concorrenza sleale ai sensi dell'art. 10-bis della Convenzione di Parigi (1967), i Membri assicurano la protezione delle informazioni segrete conformemente al paragrafo 2 e quella dei dati forniti alle autorità pubbliche o agli organismi pubblici conformemente al paragrafo 3.*

2. *Le persone fisiche e giuridiche hanno la facoltà di vietare che, salvo proprio consenso, le informazioni sottoposte al loro legittimo controllo siano rivelate a terzi oppure acquisite o utilizzate da parte di terzi in un modo contrario a leali pratiche commerciali nella misura in cui tali informazioni:*

a) *siano segrete nel senso che non sono, nel loro insieme o nella precisa configurazione e combinazione dei loro elementi, generalmente note o facilmente accessibili a persone che normalmente si occupano del tipo di informazioni in questione;*

b) *abbiano valore commerciale in quanto segrete; e*

c) *siano state sottoposte, da parte della persona al cui legittimo controllo sono soggette, a misure adeguate nel caso in questione intese a mantenerle segrete ..."*⁴⁴.

La norma fa riferimento all'art. 10-bis della Convenzione dell'Unione di Parigi (di seguito anche CUP), che nel suo secondo comma sancisce, attraverso una clausola generale, la slealtà concorrenziale delle condotte contrarie "agli usi onesti in materia industriale o commerciale"⁴⁵. Al suo terzo comma l'art. 10-bis indica poi un elenco aperto di fattispecie concorrenzialmente sleali, senza tuttavia che vi sia alcun riferimento alla condotta di sottrazione di informazioni segrete⁴⁶.

L'art. 39 TRIPs non fa quindi altro che completare l'elenco già contenuto nell'art. 10-bis della CUP, includendo fra gli atti di concorrenza sleale anche l'acquisizione, l'uso e/o la comunicazione a terzi, senza il consenso del titolare, delle informazioni segrete, con modalità contrarie alle oneste pratiche commerciali⁴⁷.

⁴³ L. DETERMANN - L. SCHMAUS - J. TAM, *Trade Secret Protection Measures and New Harmonized Laws*, in *The Computer and Internet Lawyer*, gennaio 2017, p. 1.

⁴⁴ Art. 39 TRIPs.

⁴⁵ Art. 10-bis, comma 2, CUP.

⁴⁶ Art. 10-bis, comma 3, CUP.

⁴⁷ G. SURBLYTE, *Enhancing TRIPs: Trade Secrets and Reverse Engineering*, H. ULLRICH - R.M. HILTY - M. LAMPING - J. DREXEL (a cura di), *TRIPs plus 20 - From Trade Rules to Market Principles*, Monaco, Springer, 2016, pp. 728-729. L'Autrice ricorda come, nel corso delle negoziazioni, alcuni rappresentanti dei Paesi rilevavano di non avere alcun mandato a negoziare un Accordo sulla tutela delle informazioni riservate, bensì su diritti di proprietà intellettuale. L'"escamotage" fu quello di includere tramite un richiamo all'art.

La maggiore problematica della disciplina sulle informazioni riservate contenuta nei TRIPs è stata l'implementazione da parte dei singoli Paesi membri. Nonostante la presenza di una chiara disciplina sulle informazioni segrete contenuta nei TRIPs, è accaduto infatti che, soprattutto a livello EU non vi fosse una disciplina uniforme. Questa è stata appunto una delle ragioni che ha portato alla decisione di emanare una Direttiva europea sulle informazioni segrete.

Ciò è ancor più evidente per la disciplina del *reverse engineering*, per cui non è stata prevista alcuna norma armonizzatrice all'interno dei TRIPs, e dunque a maggior ragione manca un'armonizzazione europea⁴⁸. Sebbene non sia stata espressamente regolata dai TRIPs, si ritiene tuttavia che se effettuata conformemente alle leali pratiche di commercio l'attività di *reverse engineering* sia ammissibile e lecita⁴⁹.

Il vantaggio di mantenere un'informazione segreta rispetto alle altre forme di tutela IP, tutte temporalmente limitate, è quella di avere una tutela potenzialmente illimitata nel tempo che verrà meno solo quando verranno meno i presupposti necessari per considerare un'informazione come segreta.

In Europa, sino all'emissione della Direttiva sulle informazioni segrete, non vi era una disciplina europea uniforme, bensì singole normative nazionali. Come emerso da uno studio commissionato nel 2013 dall'Unione Europea a livello europeo, nonostante la presenza dei TRIPs, non vi è un'armonizzazione né sulla definizione di informazioni segrete né sulla tutela fornita alle medesime⁵⁰. La Direttiva dovrà essere implementata da tutti gli Stati europei entro il 9 giugno 2018.

La Direttiva prevede una definizione di informazione segreta che costituisce una "trasposizione" dell'art. 39 TRIPs nell'ordinamento europeo⁵¹.

10-bis della Convenzione di Parigi la norma sulle informazioni segrete nell'ambito della tutela sulla concorrenza sleale.

⁴⁸ G. SURBLYTE, cit. (nt. 51), pp.730-733 e 740.

⁴⁹ C. GALLI, *Le limitazioni dell'esclusiva brevettuale*, C. GALLI -A.M. GAMBINO (a cura di), *Codice commentato della proprietà industriale e intellettuale*, Utet Giuridica, Torino, 2011, p. 904.

⁵⁰ *Study on Trade Secrets and Confidential Business Information in the Internal Market*, in http://ec.europa.eu/internal_market/iprenforcement/docs/trade-secrets/130711_final-study_en.pdf.

⁵¹ L. DETERMANN - L. SCHMAUS L. - J. TAM, cit. (nt. 47), p. 4; G. CHIAPPETTA, *La Proposta di direttiva e le linee di evoluzione della disciplina del Know-how rectius delle informazioni aziendali riservate*, in *Il Diritto Industriale*, 2016, 2, pp. 169-171. Si riporta qui di seguito l'art. 2, comma 1, Direttiva sulle informazioni segrete: "«segreto commerciale», informazioni che soddisfano tutti i seguenti requisiti: a) sono segrete nel senso che non sono, nel loro insieme o nella precisa configurazione e combinazione dei loro elementi, generalmente note o facilmente accessibili a persone che normalmente si occupano del tipo di informazioni in questione; b) hanno valore commerciale in quanto segrete; c) sono state sottoposte a misure ragionevoli, secondo le circostanze, da parte della persona al cui legittimo controllo sono soggette, a mantenerle segrete".

Per quanto attiene la legittimità di attività di *reverse engineering*, questo è ammesso dall'art. 3, comma 1, lett. b)⁵², che, tuttavia dovrà essere coordinato con il Considerando 39, secondo cui “*La presente direttiva non dovrebbe pregiudicare l'applicazione di qualsiasi altra pertinente normativa in altri settori, compresi i diritti di proprietà intellettuale, e il diritto contrattuale. Tuttavia, se l'ambito di applicazione della direttiva 2004/48/CE del Parlamento europeo e del Consiglio e quello della presente direttiva si sovrappongono, la presente direttiva prevale in quanto lex specialis*”⁵³.

Proprio alla luce del combinato disposto delle suddette norme, come scritto anche nel *Position Paper* del *Max Planck Institute* sulla Proposta di Direttiva sulle informazioni segrete, la norma relativa all'ammissibilità delle attività di *reverse engineering* rimane “*largely ineffective*” per quanto attiene la regolamentazione dell'attività di *reverse engineering* sul software, regolata dalla speciale disciplina contenuta nella Direttiva Software⁵⁴.

I commenti del *Max Planck Institute* alla Direttiva sono peraltro chiari nel confermare che la legittimità dell'acquisizione delle informazioni riservate attraverso l'attività di *reverse engineering* comporta anche la legittimità del loro utilizzo susseguente. Questo è un punto problematico soprattutto laddove non vi siano altri strumenti di tutela nonostante gli ingenti investimenti effettuati sul prodotto sul quale viene poi effettuata l'attività di ingegneria inversa⁵⁵.

L'art. 1 della Direttiva sulle informazioni segrete indica i margini *in melius* entro i quali i legislatori nazionali possono implementare la Direttiva medesima: ad esempio, sarà possibile elevare, come già accaduto da anni in Italia con gli artt. 98 e 99 del Codice di proprietà industriale (c.p.i.), il diritto sulle informazioni segrete a livello di vero e proprio diritto di proprietà industriale – con conseguente possibilità di avere sistemi eterogenei (con buona pace del legislatore europeo e dell'obiettivo di uniformità) di tutela fra chi li inserirà nella fattispecie della concorrenza

⁵² Art. 3, comma 1, lett. b), Direttiva sulle informazioni segrete: “1. L'acquisizione di un segreto commerciale è considerata lecita qualora il segreto commerciale sia ottenuto con una delle seguenti modalità:... b) osservazione, studio, smontaggio o prova di un prodotto o di un oggetto messo a disposizione del pubblico o lecitamente in possesso del soggetto che acquisisce le informazioni, il quale è libero da qualsiasi obbligo giuridicamente valido di imporre restrizioni all'acquisizione del segreto commerciale”.

⁵³ Considerando 39 Direttiva sulle informazioni segrete.

⁵⁴ Si vedano i *Comments of the Max Planck Institute for Innovation and Competition of 3 June 2014 on the Proposal of the European Commission for a Directive on the protection of undisclosed know-how and business information (trade secrets) against their unlawful acquisition, use and disclosure of 28 November 2013, COM(2013) 813 final, in http://www.ip.mpg.de/fileadmin/ipmpg/content/stellungnahmen/translation_stellungnahme_tsp_mpi_clear_af_with_changes_01.pdf.*

⁵⁵ Si vedano ancora i citati *Comments of the Max Planck Institute for Innovation and Competition of 3 June 2014*. Arrivano alle medesime conclusioni anche L. DETERMANN - L. SCHMAUS - J. TAM, cit. (nt. 47), p. 5.

sleale e chi li “eleverà” a diritti assoluti tutelati anche nei confronti di terzi –, in “deroga” alla Direttiva che, analogamente ai TRIPs, non mira a creare un diritto esclusivo⁵⁶.

5. *Osservazioni sul coordinamento degli artt. 5 e 6 della Direttiva Software con la Direttiva sulle informazioni segrete*

Come si è già accennato in precedenza, l’impiego “secondario” delle informazioni ricavate legittimamente dall’attività di *reverse engineering* è ben regolamentata dal comma 2 dell’art. 6 della Direttiva Software per quanto attiene l’attività di decompilazione del software ai fini dell’ottenimento dell’interoperabilità.

In particolare, non essendo la definizione di informazione di cui alla Direttiva Software, circoscritta alle sole informazioni tutelabili con il diritto d’autore, si potrebbe ottenere una tutela indiretta anche delle informazioni segrete sottese al programma, ivi comprese le idee e i principi che in teoria non dovrebbero rientrare nella tutela autoriale predisposta dalla Direttiva⁵⁷.

Per quanto attiene le informazioni ottenute in conformità all’art. 6, comma 1, Direttiva Software, vengono previsti dei limiti al loro sfruttamento, riconducibili ad una forma di tutela *sui generis*. In particolare, un’utilizzazione delle informazioni ottenute conformemente al primo comma, che non rispetti i limiti imposti dalla lett. a) e/o dalla lett. b) del secondo comma – fatto salvo quindi l’uso in violazione della lett. c) che come visto nel capitolo precedente costituisce una vera e propria violazione del diritto d’autore –, costituirà una violazione delle norme in materia di informazioni segrete e/o condotta concorrenzialmente sleale, nonché inadempimento contrattuale qualora vi siano dei rapporti contrattuali⁵⁸.

In assenza di una norma analoga al secondo comma dell’art. 6 della Direttiva Software per le informazioni ottenute dalle attività di manutenzione correttiva e/o dalla *black box analysis* di cui all’art. 5 della Direttiva Software, si può ritenere applicabile alle medesime, qualora ne ricorrano le condizioni, la normativa sulle informazioni segrete, laddove tali informazioni non siano già state trasformate in codice sorgente: infatti, nel caso in cui le suddette informazioni vengano implementate in un programma, tornerà ad essere applicabile la disciplina di cui alla Direttiva Software.

Un dubbio sull’esatta linea di demarcazione fra tutela come informazione segreta e tutela autoriale è tuttavia presente, in quanto la

⁵⁶ G. CHIAPPETTA, cit. (nt. 55), p. 171.

⁵⁷ W. BLOCHER - M. WALTER, cit. (nt. 38), p. 180.

⁵⁸ G. GUGLIELMETTI, cit. (nt. 10), pp. 192-196; E. DOTI, *I diritti degli utenti legittimi*, A. MUSSO (a cura di), *Il software tra brevetto e diritto d’autore. Regime “proprietario” o open source?*, Bononia University Press, Bologna, 2005, p. 151.

Direttiva Software ricomprende nel concetto di programma per elaboratore anche i “lavori preparatori” (come si è visto l’interpretazione del termine “lavori preparatori” non è chiara nella Direttiva Software e, ad oggi, non vi sono state decisioni a livello europeo che abbiano fornito un’interpretazione precisa di tale concetto: la valutazione dovrà quindi essere effettuata caso per caso).

Si potrebbe quindi eventualmente far rientrare nell’ambito della tutela autoriale l’informazione ottenuta legittimamente che andrà presumibilmente a costituire parte dei lavori preparatori del programma che verrà in futuro sviluppato. Tuttavia, dal momento in cui tali informazioni sono state ottenute legittimamente, si potrebbe ritenere che siano comunque al di fuori degli elementi tutelati e tutelabili dal diritto d’autore. Ciò vale, ad esempio, per gli algoritmi o per le specifiche sulle interfacce che non rientrano nell’ambito della tutela autoriale.

Si potrebbe quindi ipotizzare una tutela come informazioni segrete delle specifiche sulle interfacce o degli algoritmi, sino a quando i medesimi non vengano implementate in un codice (o in un progetto di codice riconducibile nell’ambito della categoria del “materiale preparatorio”), ed allora alla tutela come informazione segreta si sovrapporrà, prevalendo, la già esaminata tutela autoriale prevista per il software.

Il tema della tutelabilità e dei limiti di riutilizzo delle informazioni ottenute attraverso la c.d. *black box analysis* non è peraltro trascurabile, visto il trend verso una sempre maggiore diffusione dei programmi offerti “*as a service*” (SaaS), in cui, come si è già ricordato, il codice oggetto non è visibile all’utente finale che fruisce del programma senza tuttavia avere accesso diretto al codice oggetto (contenuto in un server a cui ha accesso solo il programmatore), e conseguentemente non decompilabile: per i SaaS ci si può quindi limitare solo ad un’analisi del funzionamento attraverso la *black box analysis*.

Ovviamente quanto detto vale sino a quando non interviene la tutela brevettuale che introduce un’altra possibile tutela del software sovrapponibile a quelle già esaminate.

6. Conclusioni

Sin dagli albori dell’industria software, lo strumento della segretezza è stato lo strumento principe utilizzato dalle Software House per proteggere i propri software. Infatti, a prescindere dalla tipologia di tutela che si fornisce al software, questo strumento è di fatto presente nel mondo software che segue il c.d. *close code model*: il programma viene commercializzato come codice oggetto, ed il programmatore si guarda bene dal diffondere il codice sorgente che rimane quindi segreto.

Pertanto, nonostante l’aver ricondotto il software alla tutela autoriale – ma lo stesso discorso può essere fatto, in parte, anche per la

tutela brevettuale delle c.d. *software related invention* –, l'aspetto di segretezza continua a riemergere laddove la disciplina autoriale presenti dei "bug" (per usare una terminologia che piace agli informatici), o comunque dei limiti di tutela, non potendosi la tutela autoriale, ad esempio, estendersi ai principi e alle idee.

E la Direttiva sulle informazioni segrete non ha fatto altro che mettere in luce le "anomalie" della tutela del software attraverso il diritto d'autore: basti pensare ai limiti imposti dalla Direttiva Software all'attività di *reverse engineering*, che costituisce evidentemente una distorsione della *ratio* originaria alla base del *reverse engineering*. Infatti, il *reverse engineering*, originariamente nato nel mondo delle informazioni segrete (ma oggi preso in prestito, sia dal diritto d'autore, sia, di recente, anche dal diritto dei brevetti con l'Accordo sulla Unified Patent Court), consistente, da una parte, nel far progredire la società, dall'altra, nel consentire al titolare delle informazioni segrete, nel corso del tempo che il terzo impiega per risalire alle medesime informazioni attraverso le attività di *reverse engineering*, il recupero degli investimenti effettuati nello sviluppo del *know-how*⁵⁹. Tale *ratio* viene evidentemente distorta nel sistema costruito dalla Direttiva Software.

⁵⁹ P. SAMULESON – S. SCOTCHMER, *The Law and Economics of reverse engineering*, in *The Yale Law Journal*, 2002, III, pp. 1585 e ss.