

CAPITOLO PRIMO

Il bene oggetto del contratto

SOMMARIO: 1. Inquadramento normativo. – 2. Inquadramento tecnico. – 2.1. In particolare, codice sorgente e codice oggetto. – 2.2. Compilazione e decompilazione. – 3. La tutela giuridica complementare *inter partes*. – 4. L'autotutela tecnologica *erga omnes*. – 5. Ai confini tra brevetto e diritto d'autore. – 6. Il software nella giurisprudenza prima del d.lg. 518/1992. – 7. Il software secondo il legislatore italiano. – 7.1. Il Registro Pubblico Speciale per i programmi per elaboratori della Società Italiana degli Autori ed Editori. – 7.2. Le recenti vicende del Bollino SIAE.

1. Inquadramento normativo

Legislazione c.c. 2575 – l. 633/1941; 48/2008; 248/2000 – d.lg. 518/1992; 205/1996; 68/2003; 140/2006 – d.p.c.m. 31/2009 – dir. 91/250/CEE; 2004/48/CE.

Bibliografia Borruso 1988 – De Sanctis 2000.

Il *software*, secondo la definizione elaborata dall'Organizzazione Mondiale della Proprietà Intellettuale, durante la riunione tenutasi a Canberra dal 2 al 6 aprile 1984, è

«espressione di un insieme organizzato e strutturato di istruzioni in qualsiasi forma o su qualunque supporto capace, direttamente o indirettamente, di far eseguire o far ottenere una funzione o un compito o far ottenere un risultato particolare per mezzo di un sistema di elaborazione elettronica dell'informazione».

(WIPO – World Intellectual Property Organization, 1984).

La formulazione, volutamente ampia, è in grado di comprendere ogni possibile forma di *software* e ben si adatta al rapido evolversi della scienza informatica. Autorevoli Voci, partendo dalla suddetta definizione, ne hanno proposto una più sintetica, secondo la quale il programma è il

«complesso di tutte le istruzioni necessarie a far eseguire al computer un determinato lavoro».

(Borruso 1988, 94).

ovvero la

«sequenza di frasi univocamente interpretabili rivolte a un calcolatore affinché ponga in essere gli enunciati».

(De Sanctis 2000, 3).

Il concetto di programma per elaboratore è stato per lungo tempo oggetto di attenzione, relativamente alle forme di tutela ad esso ricollegabili.

«Non sono proteggibili sulla base del diritto d'autore i programmi per elaboratore elettronico perché oggetto di tale protezione è pur sempre un quid destinato alla comunicazione fra esseri umani, laddove il programma, anche se consacrato in uno scritto, è destinato a fornire alla macchina una serie concatenata di istruzioni al fine di farle svolgere un preciso compito o al fine di consentirle di conseguire un determinato risultato».

(Trib. Milano 20.6.1988, *NGCC*, 1989, I, 360).

Le principali perplessità, sul regime di protezione da accordare a questo prodotto dell'intelletto umano, originavano soprattutto dall'immaterialità e dalle peculiarità tecniche che lo distinguono dalle altre opere dell'ingegno tradizionalmente intese. Da qui, il dubbio di una protezione mediante il diritto d'autore ovvero attraverso la disciplina protettiva dei brevetti.

«I programmi per elaboratore debbono qualificarsi opere dell'ingegno a tutti gli effetti».

(Pret. Cassano d'Adda 19.10.1988, *DInf*, 1989, 209).

«Non sono proteggibili sulla base del diritto d'autore i programmi per elaboratore elettronico perché oggetto di tale protezione è pur sempre un quid destinato alla comunicazione fra esseri umani, laddove il programma, anche se consacrato in uno scritto, è destinato a fornire alla macchina una serie concatenata di istruzioni al fine di farle svolgere un preciso compito o al fine di consentirle di conseguire un determinato risultato».

(Trib. Milano 20.6.1988, *NGCC*, 1989, I, 360).

In particolare, per quanto concerne la possibilità di tutelare i programmi per elaboratore mediante la disciplina brevettuale, va immediatamente rilevato che il requisito di industrialità non è sempre presente nel *software* (Sena 1994; Frassi 1997).

Il dibattito, dottrinale e giurisprudenziale, ha avuto origine negli anni sessanta, ma la necessità di proteggere il *software* da copie abusive e contraffazioni si è avvertita dopo oltre un decennio, in seguito alla commer-

cializzazione di massa dei primi *software* pacchettizzati. Fino a quel momento, il programma per l'elaboratore elettronico era un tutt'uno con la parte *hardware*; chi produceva le componenti "rigide" realizzava, da sé, anche il *software*. In siffatto contesto, inoltre, i computer erano, tra loro, incompatibili con conseguente impossibilità di riuso del *software* su piattaforme con caratteristiche diverse da quelle originarie.

Il successivo sviluppo autonomo dell'industria del *software*, rispetto a quella dell'*hardware*, e la dimensione economico-sociale degli interessi coinvolti ha creato la necessità di salvaguardare i risultati di una ricerca costosa e complessa e nel contempo di determinare correttamente l'inquadramento giuridico dei programmi per elaboratore e la relativa protezione dei diritti connessi. La ricerca di un'adeguata tutela per i produttori di programmi per elaboratore elettronico perseguiva, altresì, l'ulteriore finalità di incentivare la ricerca tecnologica che, in assenza di un'adeguata normativa, sarebbe stata pregiudicata.

I primi ad aprire la strada verso una tutela certa ed organica dei programmi per elaboratore furono gli Stati Uniti d'America, i quali, nel 1980, emanarono il *Computer Software Amendment Act* che ha condizionato le altre discipline nazionali e soprattutto Europee.

Nel frattempo, anche in Italia si maturava la necessità di avere nuove regole certe e precise finalizzate alla tutela dei prodotti *software*. In tal senso,

«La tutela *ex lege* n. 633 del 1941 dei programmi per elaboratore quali opere dell'ingegno appartenenti *lato sensu* alle scienze pur se non del tutto appagante – in considerazione del fatto che il diritto d'autore protegge la forma esterna ma non il contenuto ed inoltre il linguaggio utilizzato dal programma non è rivolto all'uomo bensì alla macchina – merita di essere perseguita in attesa di un auspicabile intervento del legislatore».

(Pret. Roma 4.5.1988, *DInf*, 1989, 555).

La strada scelta dal legislatore statunitense fu quella del diritto d'autore (o *copyright approach*); orientamento seguito anche dalla Comunità Economica Europea con la direttiva 91/250/CEE, recepita nell'ordinamento giuridico italiano attraverso il d.lg. 518 emanato il 29.12.1992 che ha novellato la l. 633/1941 (di seguito anche LdA) sul diritto d'autore.

Successivamente il legislatore italiano intervenne più volte, su singoli aspetti della disciplina, con significative novelle. Tra i principali interventi successivi al recepimento della direttiva 91/250/CEE, si ricordano: il d.p.c.m. 3.1.1994, n. 244, che ha stabilito le modalità di tenuta del Registro pubblico speciale dei programmi per elaboratore; il d.lg. 15.3.1996, n. 205, modificativo del d.lg. 29.12.1992, n. 518, in materia di tutela giu-

ridica dei programmi per elaboratore; la l. 18.8.2000, n. 248 (Nuove norme di tutela del diritto di autore); e, ancora, il d.lg. 68/2003 (sull'armonizzazione di taluni aspetti del diritto d'autore e dei diritti connessi nella società dell'informazione), il d.lg. 140/2006 (Attuazione della direttiva 2004/48/CE, cosiddetta "Direttiva *Enforcement*", sul rispetto dei diritti di proprietà intellettuale), la l. 9.1.2008 (Disposizioni concernenti la Società italiana degli autori ed editori), n. 2, la l. 18.3.2008, n. 48 (Ratifica ed esecuzione della Convenzione del Consiglio d'Europa sulla criminalità informatica, fatta a Budapest il 23.11.2001, e norme di adeguamento dell'ordinamento interno).

2. *Inquadramento tecnico*

Bibliografia Leggio 1992 – Borruso 2001 – Curtin, Foley, Sen, Morin 2005.

Il *software* è una delle due parti fondamentali del sistema elettronico avente la funzione di elaborare le informazioni ricevute e di restituire all'utente i risultati desiderati (Leggio 1992, 717 ss.); vi è, poi, la parte complementare, denominata *hardware*, che indica la struttura fisica dei dispositivi informatici ed è costituito da componenti elettronici che svolgono specifiche funzioni nel trattamento e nella trasmissione dell'informazione. Fanno parte dell'*hardware* le periferiche quali ad esempio i *monitors*, le tastiere e le stampanti.

Il *software* svolge le sue funzioni a livello logico; esso è determinato da un insieme di istruzioni che consentono all'*hardware* di svolgere i propri compiti. Curiosa, nel delineare la distinzione tra le due parti fondamentali del sistema informatico, la Voce secondo la quale

«l'*hardware* è la parte che puoi prendere a calci; il *software* quella contro cui puoi solo imprecare».

(Curtin, Foley, Sen, Morin 2005, 20).

Le istruzioni che compongono il *software* sono scritte in una particolare lingua denominata linguaggio di programmazione. A voler essere precisi, non esiste un unico linguaggio di programmazione ma svariate tipologie tutte dotate di una propria sintassi, proprie parole, propri segni e punteggiature, nonché una propria grammatica.

Le proposizioni linguistiche, elaborate mediante l'uso di un linguaggio di programmazione, sono le istruzioni che, secondo una specifica successione aderente all'algoritmo su cui si basa il programmatore, vanno a comporre il codice sorgente del quale si tratterà nel prosieguo.

Per ora è sufficiente precisare che, seconda delle funzioni svolte, e pertanto delle istruzioni impartite (e contenute nel codice sorgente), i programmi per elaboratore si distinguono in *software* di base, il quale viene solitamente fornito a corredo dell'*hardware* e garantisce l'esecuzione delle funzioni di accensione, spegnimento e le altre funzioni di base, e *software* applicativo, che

«serve a far sì che il computer svolga, tra i tanti lavori possibili, un lavoro particolare».

(Borruso 2001, 201 ss.).

i supporti sui quali vengono memorizzati sono le cosiddette memorie di massa (*floppy disk, hard disk, CD Rom, DVD Rom*).

2.1. In particolare, codice sorgente e codice oggetto

Bibliografia Ausiello 1986 – Falzone 1986 – Rossello 1986 – Brock 1990 – Cavani 1994 – Ciotti, Roncaglia 2003 – Lemley, Menell, Merges, Samuelson 2003.

Tralasciando la suddetta distinzione tra programmi di base e applicativi, al fine di inquadrare correttamente il *software* tra le opere dell'ingegno, è opportuno distinguere tra codice sorgente e codice oggetto. Qualunque programma ha alla sua base un listato, il quale rappresenta l'espressione creativa dell'intelletto umano. Codice sorgente e codice oggetto rappresentano due diversi modi, seppur strettamente connessi, di intendere il *software* ed, a loro volta, si distinguono per vari aspetti (Ausiello 1986, 643 ss.).

Il codice sorgente è definibile come l'insieme di istruzioni scritte, in linguaggio formale ad alto livello (C, C++, Java, python, Basic, Pascal, COBOL, ecc.), all'interno di uno o più file, le quali, considerate nel loro complesso, costituiscono il programma. Tale codice, affinché sia eseguibile dal processore, deve essere tradotto in codice oggetto (o codice macchina) attraverso un compilatore; oppure elaborato da un programma (interprete), che contemporaneamente trasforma in linguaggio macchina e memorizza su appositi *file* il codice oggetto, eseguito simultaneamente da parte del processore (Lemley, Menell, Merges, Samuelson 2003, 25 ss.; Ciotti, Roncaglia 2003, 92 ss.).

«Ed invero, senza necessità di ripercorrere tutti gli itinerari della dottrina e della giurisprudenza nell'individuazione dei caratteri dell'originalità e della creatività, basandoci anche sulla Relazione al progetto di direttiva del 1989,

secondo le indicazioni di un chiaro Autore, si può affermare che in considerazione delle peculiarità del programma per elaboratore il requisito dell'originalità è quello interpretato in chiave soggettiva nel senso che è sufficiente per il riconoscimento della tutela la circostanza che l'opera sia frutto di uno sforzo intellettuale indipendente e che non sia, pertanto, la riproduzione di quanto già realizzato da altri.

Peraltro la locuzione "in qualsiasi forma espressi" correlata ai programmi per elaboratori richiama la distinzione fra programmi – sorgente, espressi in un linguaggio della programmazione e percepibili dai tecnici, e programmi – oggetto, formulati in un linguaggio comprensibile unicamente dalla macchina, sicché, ove si consideri che il termine programma comprende anche il materiale preparatorio per il programma stesso, anche a non voler includere nella tutela penale il materiale che compone le varie fasi della creazione del programma, detta precisazione dimostra come debba essere protetta in sede penale pure la duplicazione parziale.

Né, in tal modo, si procede ad un'interpretazione analogica oppure ad ampliare la fattispecie penale in contrasto con il principio di tassatività, giacché, invece, si riduce l'ambito del precetto che potrebbe essere esteso al materiale preparatorio, secondo la definizione giuridica su individuata, e, nel contempo, si adegua la norma ai contenuti tecnici del *software* senza, però, accedere ad una concezione minimalista di originalità e creatività e senza estendere la protezione ad ogni singola fase non dotata di propria autonomia e non costituente il nucleo fondamentale del programma.

Infatti, il *software* è caratterizzato, nella concezione della creatività, dall'essere un ritrovato del progresso tecnologico, diretto alla produzione di un risultato utile, oltre che una creazione intellettuale, sicché sconta del primo aspetto una diminuzione del concetto di originalità, perché l'innovazione tecnologica non è completamente astratta dal precedente, ma costituisce spesso elaborazione, adeguamento e perfezionamento.

Pertanto occorre graduare i vari contenuti dell'atto di elaborazione creativa di dati reali, istruzioni tecniche o idee, seguendo in ordine all'individuazione di un contenuto minimo di originalità i criteri elaborati dalla dottrina e dalla giurisprudenza con riguardo alle opere di natura tecnico-scientifica in modo tale da apprestare la tutela penale più al programma – sorgente ed al *code line* invece che al programma – oggetto, qualora lo sforzo creativo addizionale sia minimo.

Ed invero la formula contenuta nella direttiva comunitaria secondo cui un programma per elaboratore è tutelato se è il risultato della creazione originale dell'autore comporta che esso sia frutto di lavoro autonomo e non di mera riproduzione o duplicazione anche parziale».

(Cass. pen. Sez. III, 24.4.2002, n. 15509).

L'operazione attraverso la quale si trasforma il codice sorgente in codice oggetto è chiamata compilazione ed avviene tramite appositi *software* detti "interpreti" oppure "compilatori" (Brock 1990, 1, 421 ss.; Rossello 1986, 208).

I linguaggi macchina, sono binari, e quindi composti da istruzioni con sequenza numerica zero (0) e uno (1); per questo motivo l'immediato prodotto della mente umana è, principalmente, il codice scritto in linguaggio di alto livello (*HLL* o *High Level Language*) avente una specifica sintassi organizzata in proposizioni logiche. I linguaggi HLL permettono la scrittura di programmi potenzialmente eseguibili su qualunque calcolatore, in quanto essi descrivono un algoritmo mediante una sequenza di istruzioni (o proposizioni) tipiche del linguaggio utilizzato (Falzone 1986, 367 ss.).

La principale differenza tra codice sorgente e codice oggetto è, pertanto, costituita dal fatto che quest'ultimo è direttamente eseguibile dall'elaboratore, in quanto compilato in linguaggio macchina leggibile direttamente dal processore. Può, così, affermarsi che il codice sorgente sta al programmatore come il codice oggetto sta all'elaboratore; il primo è comprensibile dall'uomo, il secondo direttamente eseguibile dal calcolatore.

A questa distinzione di carattere prettamente "tecnico" corrisponde altrettanta distinzione in termini di tutela giuridica; è stato infatti osservato che

«Il codice sorgente è un'opera dell'ingegno diversa e protetta separatamente dal relativo programma esecutivo, onde il contratto di licenza di *software* esecutivo non si estende al relativo codice sorgente, in difetto di specifica pattuizione delle parti; onde non può essere accolta la domanda di risoluzione (per inadempimento o per eccessiva onerosità sopravvenuta) di un contratto di licenza di programma esecutivo, per mancata fornitura del codice da parte del licenziante, quando la licenza non si estenda al codice sorgente».

(Trib. Roma 31.1.2001, *AIDA*, 2000, Rep. I, 22).

Il codice sorgente ed il codice oggetto sono autonomi, seppur collegati, e destinatari, ciascuno separatamente dall'altro, della tutela che l'ordinamento accorda alle opere dell'ingegno. Il primo è direttamente collegabile al frutto dell'ingegno umano, il codice oggetto, sebbene non intelligibile dall'uomo è anch'esso un prodotto dell'intelletto,

«nonostante le sue istruzioni siano indirizzate ad una macchina».

(Pret. Bari, ord. 11.2.1991, *FI*, 1991, I, 2258).

Non pare sussistano dubbi sul fatto che il legislatore abbia voluto comprendere, nella nozione di programma per elaboratore, sia il sorgente, sia l'oggetto (Cavani 1994, 14).

È, quindi, corretto affermare che entrambi possono godere della prote-

zione prevista dalla Legge sul diritto d'autore anche in modo autonomo e disgiunto. L'autore, unico *dominus* dell'opera, detta le regole di circolazione della stessa; spetta unicamente a lui la scelta di cedere anche il codice sorgente o, come capita nella maggior parte dei casi, soltanto il codice oggetto e beneficiare, in conseguenza della scelta fatta, di una tutela congiunta o disgiunta per due differenti modalità espressive della personalità umana.

«Il codice sorgente è un'opera dell'ingegno diversa e protetta separatamente dal relativo programma esecutivo, onde il contratto di licenza di *software* esecutivo non si estende al relativo codice sorgente, in difetto di specifica pattuizione delle parti; onde non può essere accolta la domanda di risoluzione (per inadempimento o per eccessiva onerosità sopravvenuta) di un contratto di licenza di programma esecutivo, per mancata fornitura del codice da parte del licenziante, quando la licenza non si estenda al codice sorgente».

(Tri. Roma 31.1.2001, *AIDA*, 2000, Rep. I, 22).

2.2. *Compilazione e decompilazione*

Legislazione l. 633/1941 64-*quater*.

Bibliografia Guglielmetti 1994 – Marzano 1998 – Ubertazzi 2007.

La compilazione è un'operazione che potrebbe definirsi, seppure non in senso assoluto, a senso unico ma non è impossibile il procedimento inverso, che dal codice oggetto porta al codice sorgente. La contraddittorietà, di quanto appena affermato, svanisce se si precisa che in seguito alla traduzione del codice sorgente in codice oggetto, è impossibile riconvertirla esattamente nella sequenza originaria di istruzioni. Il procedimento di decompilazione, di solito, genera un listato che differisce da quello originariamente scritto dal programmatore; i compilatori (le macchine), infatti, non si limitano ad una semplice sostituzione di termini tra i diversi linguaggi, bensì effettuano anche una serie di ottimizzazioni al fine di migliorare la velocità del programma e l'utilizzazione delle risorse *hardware*. Per questo motivo, a seguito di decompilazione, il listato ottenuto non è identico all'originario codice sorgente la cui struttura (i nomi dei moduli scelti dal programmatore e la precisa sequenza delle istruzioni usate) è andata perduta nel processo di prima trasformazione in linguaggio macchina.

La LdA, all'art. 64-*quater*, si occupa del tema della decompilazione (o *reverse engineering*) (Guglielmetti 1994, 10), ossia del procedimento che, consentendo di risalire dal programma oggetto al programma sorgente

(non identico ma) equivalente a quello originariamente scritto dal programmatore, risulta necessario per soddisfare esigenze di interoperabilità tra programmi per elaboratore creati autonomamente. L'interoperabilità è

«la capacità di due o più sistemi informatici di scambiare informazioni e di usare reciprocamente le informazioni scambiate».

(Ubertazzi 2007, 175).

Lo scambio di dati tra programmi può avvenire tra *software* dello stesso tipo ma di produttori diversi, ovvero tra *software* che hanno un rapporto di complementarità ed ha un'importanza fondamentale per l'utilizzatore, al punto da costituire il fondamento di una deroga ai diritti esclusivi dell'autore. È proprio sulla necessità di garantire l'interoperabilità, infatti, che l'art. 64-*quater* della LdA accorda al soggetto, il quale è legittimato a usare una copia del programma, il potere di ricavare tutte le informazioni necessarie a garantirla. Se il legittimo licenziatario patisce un problema di interoperabilità, e soltanto in presenza di siffatta condizione, può, senza l'autorizzazione del titolare dei diritti e a condizione che quelle stesse informazioni non siano diversamente disponibili (art. 64-*quater*, 1° co., lett. b), procedere alla *reverse engineering* e, pertanto, giungere alle regole base, idee e principi che formano la struttura di base del programma.

La disposizione in esame persegue, contemporaneamente, un fine privatistico (di garantire al legittimo acquirente la piena disponibilità, ed interoperabilità, del prodotto) ed una finalità pubblicistica (di favorire lo sviluppo tecnologico del settore; quindi il progresso e la conoscenza).

Si tratta, come già precisato, di una disposizione che interrompe l'antigiuridicità di una condotta vietata; infatti, in assenza di una precisa autorizzazione del titolare dei diritti esclusivi, l'attività di decompilazione, come di ogni altra attività contraria al consenso dell'autore, è illecita.

Quando l'autore decide di rendere noto il codice sorgente, può farlo adottando un'apposita licenza *open source* per la distribuzione del proprio prodotto, ovvero, semplicemente pubblicandolo. Viceversa, in tutti i casi nei quali l'autore cela il contenuto del codice sorgente, e si limita a distribuire il codice oggetto, questo suo comportamento è da intendere quale manifestazione di volontà alla segretezza del listato. Da ciò deriva l'illiceità delle condotte di scomposizione dei passaggi logici che conducono all'idea di base che regge questo o quel *software*.

Siffatto comportamento, non è, al contrario, illecito se giustificato dalla necessità di risolvere problemi di ininteroperabilità con altri programmi ma nel rispetto dei seguenti limiti.

Le attività di decompilazione devono limitarsi alle parti del programma originale necessarie per conseguire l'interoperabilità (art. 64-*quater*,

1° co., lett. c); le informazioni ricavate dall'attività di *reverse engineering* non possono essere utilizzate a fini diversi dal conseguimento dell'interoperabilità del programma; le informazioni ed i risultati ottenuti dall'attività di analisi e de compilazione non possono essere comunicato a terzi (art. 64-*quater*, 2° co., lett. b); infine, il codice sorgente ottenuto dall'attività di decompilazione non può essere utilizzato per lo sviluppo, la produzione o la commercializzazione di un programma per elaboratore sostanzialmente simile nella sua forma espressiva, o per ogni altra attività che violi il diritto di autore (art. 64-*quater*, 2° co., lett. c). A proposito di quest'ultimo limite, la Giurisprudenza di legittimità ha rilevato che, l'art. 64-*quater*

«nel disciplinare il c.d. *reverse engineering*, considera in modo eminente gli interessi del titolare del programma, nonostante l'importanza economica dell'interoperatività dei programmi e delimita in maniera rigorosa la possibilità di riproduzione, anche parziale, vietando che le informazioni “siano utilizzate per ogni altra attività che violi il diritto d'autore”».

(Cass. pen. Sez. III, 24.4.2002, n. 15509).

Si rileva una grande differenza tra il sistema comunitario e l'ordinamento statunitense, il quale consente la decompilazione anche se semplicemente giustificata dal fine di ottenere le informazioni utili a creare un nuovo prodotto, indipendente dal primo e, quindi, senza la necessaria condizione dell'interoperabilità imposta dalla LdA italiana (Marzano 1998, 446 ss.).

3. La tutela giuridica complementare inter partes

Legislazione c.c. 1374, 2105, 2598 – l. 633/1941 – d.lg. 518/1992.

Bibliografia Carnevali 1984 – Frignani 1984 – Carnevali 1984 – Capograssi 1985 – Ghidini 1986 – Bianchi 1988 – Ristuccia, Zeno Zencovich 1990 – Guglielmetti 1994 – Ristuccia, Zeno Zencovich 1993 – D'Aiotti 1994 – Rinaldi 1994 – Bertani 1998.

Le definizioni di *software* coniate dalla dottrina e dalla giurisprudenza, nel corso degli anni, sono tutte concordi nell'annoverarlo tra i beni giuridici appartenenti alla categoria delle creazioni intellettuali (Capograssi 1985, 21; Ristuccia, Zeno Zencovich 1990, 55; Frignani 1984, 57).

In effetti, le caratteristiche proprie del *software*, lo rendono diverso rispetto alle comuni opere dell'ingegno; si tratta di un prodotto immateriale e inconsumabile, tale da poter essere utilizzato contemporaneamente da un numero indefinito di utenti, senza che ne diminuisca l'utilità intrinseca.

Al pari delle altre creazioni intellettuali (invenzioni, modelli industriali, opere dell'ingegno), il *software* deve essere ricondotto al concetto di bene autonomo in grado di assicurare, tanto il diritto di paternità all'autore, che quello di utilizzazione o di sfruttamento economico dell'opera stessa all'utilizzatore. Si tratta di diritti assoluti, tali da poter essere fatti valere nei confronti di chiunque.

Per lungo tempo, e fino a quando il legislatore non ha manifestato una scelta precisa (d.lg. 518/1992), la dottrina e la giurisprudenza hanno cercato di ricondurre il *software* verso la tutela brevettuale ovvero verso la LdA (i tratti peculiari di questo percorso interpretativo saranno illustrati nei seguenti paragrafi).

Entrambe le soluzioni sono in grado di offrire una tutela *erga omnes*; ma è altresì vero che il bene immateriale *software* (e quindi il suo autore) può godere di alcuni strumenti di tutela *inter partes*, la cui valenza è complementare, e meramente rafforzativa. Si tratta di rimedi azionabili, per certi versi, esclusivamente nei confronti di soggetti che hanno un rapporto negoziale (ad esempio un contratto di licenza d'uso) con il titolare dei diritti sull'opera, per altri, in presenza di specifiche condizioni (per esempio qualità professionali dei soggetti coinvolti).

Per meglio comprendere la portata di quanto pocanzi accennato, si consideri, per il primo tipo, la tutela contrattuale attraverso l'inserimento di clausole che regolino e/o limitino l'uso da parte degli utenti; quale seconda tipologia, l'applicazione delle norme in tema di concorrenza sleale.

Così, il contratto che concede l'uso del *software* può contenere clausole quale il divieto di fare copie del programma, il divieto di trasferirlo a terzi, il divieto di divulgarne il contenuto, e altre simili.

La pratica offre svariate possibilità in tal senso ed, in alcuni casi, gli autori fanno un uso sproporzionato delle clausole di protezione, le quali finiscono per avere un contenuto vessatorio e, pertanto, inefficace o, addirittura, nullo. Si tratta di limitazioni pattizie della garanzia e della responsabilità del produttore, di cui meglio si dirà nel prosieguo della presente opera.

In questa sede, ci si vuole, invece limitare a descrivere l'uso "bonario" di clausole negoziali protettive per l'autore, che lo tutelano da abusi che potrebbero compiere i licenziatari (contratto di licenza d'uso) ovvero gli sviluppatori (contratto di sviluppo) del programma per elaboratore.

Un ipotetico vincolo negoziale del primo tipo è quello che vieta l'installazione contemporanea del prodotto su più macchine da parte del licenziatario che acquista con licenza, cosiddetta, monouso. In siffatta ipotesi, come in altre simili, il *software* è l'oggetto dell'obbligo di non fare (non installare), con valore esclusivamente *inter partes*; di conseguenza,

alla violazione dell'obbligo (installazione contemporanea su più macchine), non consegue una tutela assoluta. L'autore (o, più in generale, il licenziante) potrà invocare i rimedi contrattuali nei confronti della controparte ma non potrà impedire ai terzi quel comportamento inibito, al licenziante, mediante una specifica clausola contrattuale.

«La clausola di un contratto di licenza di distribuzione di un programma per elaboratore recante il divieto per il distributore di commercializzare il prodotto se non ad "utenti finali", può essere opposta al terzo acquirente "non utente" finale estraneo al contratto di distribuzione. Trattasi dunque di una eccezione al principio dell'esaurimento del diritto patrimoniale d'autore, posto che si ammette un diritto dell'autore del programma di controllare, anche successivamente al primo trasferimento, l'ulteriore locazione, licenza o vendita del programma o di copia dello stesso».

(Pret. Monza 27.9.1989, *DInf*, 1991, 924).

Una specifica clausola contrattuale, se non rispettata, costituisce la fonte dell'obbligazione risarcitoria da inadempimento e legittima l'autore (o più genericamente l'alienante, titolare dei diritti patrimoniali sull'opera) ad invocare esperire l'apposita azione a propria tutela.

Vi sono altri casi in cui le limitazioni contrattuali sono imposte direttamente dalla legge, seppur non previsti espressamente nella singola pattuizione. Si tratta della cosiddetta integrazione del contratto (art. 1374 c.c.), per la quale il contratto vincola le parti non soltanto a quanto in esso è stabilito, ma anche a quanto disposto (o imposto) dalla legge o, in mancanza, dagli usi e dall'equità.

Ebbene, anche i contratti del *software*, mediante le disposizioni di diritto comune sono integrati nel loro contenuto ed, in tal senso, limitano alcune libertà delle parti. Un chiaro esempio, che tutela il datore di lavoro (titolare dei diritti patrimoniali sull'opera) limitando lo sviluppatore, è quello contenuto nell'art. 2105 c.c. in tema di obbligo di fedeltà nel rapporto di lavoro subordinato (Carnevali 1984, 251 ss.; Ghidini 1986, 325; Guglielmetti 1994, 162-165).

Ancora una volta, come nell'esempio precedente, il *software* è l'oggetto dell'obbligo di non fare (non rivelare informazioni sul prodotto), con valore esclusivamente *inter partes*; se il lavoratore diffonde ingiustificatamente il codice sorgente, violando, di fatto, il segreto su aspetti organizzativi dell'impresa, può incorrere in sanzioni disciplinari (Bianchi 1988, 1 ss.); tutela non azionabile nei confronti di soggetti terzi, estranei al rapporto di lavoro.

Altra possibile tutela potrebbe aversi con il ricorso alle disposizioni codicistiche dettate in tema di concorrenza sleale ed in particolare l'art.

2598 c.c. (Carnevali 1984, 251 ss.; Ristuccia, Zeno Zencovich 1993, 56 ss.; D'Aiotti 1994, 222-223; Rinaldi 1994, 271).

Nello specifico, la disposizione in esame prevede, al n. 1, tre distinte condotte, tutte potenzialmente riferibili al *software*. La prima consiste nell'uso di

«nomi o segni distintivi idonei a produrre confusione con i nomi o i segni distintivi legittimamente usati da altri [...]».

(Art. 2598, n. 1, c.c.),

la seconda, di particolare interesse per i programmi per elaboratore, definisce come concorrenza sleale l'imitazione servile dei «*prodotti di un concorrente*». Il *software* è, infatti, un prodotto facilmente imitabile, soprattutto nel suo aspetto esterno (per esempio l'interfaccia) (Ristuccia, Zeno Zencovich 1993, 58). Al contrario, l'imitazione delle funzioni interne di un *software* è, quasi mai, in grado di ingenerare confusione sui programmi interessati.

Secondo la Giurisprudenza di merito

«le caratteristiche intrinseche del *software* che fa funzionare la banca dati non costituiscono di norma, agli occhi del consumatore, elemento differenziatore del prodotto; pertanto, la copia parziale della struttura e dei contenuti di un Cd-rom contenente una banca dati giuridica, edito da un imprenditore rivale, pur potendo integrare altri tipi di illecito, non costituisce imitazione servile del prodotto altrui *ex art. 2598, n. 1, c.c. [...]*».

(Trib. Roma 6.8.2001, *DA*, 2004, 374).

La terza condotta, infine, a carattere residuale, consiste nel compimento attraverso “qualsiasi altro mezzo” di “atti idonei a creare confusione con i prodotti e con l'attività di un concorrente”.

Un riferimento più generale alla concorrenza sleale ha condotto la Giurisprudenza (Pret. Bari 11.2.1991; Pret. Roma 4.7.1988; Pret. Roma ord. 25.5.1982) a ritenere applicabile anche il n. 3 dell'art. 2598 c.c., il quale considera l'attività di chi si vale

«direttamente o indirettamente di ogni altro mezzo non conforme ai principi della correttezza professionale e idoneo a danneggiare l'altrui azienda».

(Art. 2598, n. 1, c.c.).

Appaiono, pertanto, riconducibili alla suddetta previsione normativa tutti gli atti di riproduzione che consentono ad un imprenditore di risparmiare le spese di ricerca e di programmazione. In tal senso conforme si è

pronunciata, ancora, la giurisprudenza di merito sopracitata, sostenendo che

«[...] La riscontrata pedissequa riproduzione dell'altrui *software*, relativo alla banca dati integra l'illecito previsto dal n. 3 dello stesso art. 2598 c.c., ove l'ex dipendente dell'imprenditore danneggiato abbia passato all'imprenditore slealmente concorrente informazioni riservate: in tal caso l'ex dipendente, sebbene sia soggetto non imprenditore, è responsabile solidalmente dell'illecito concorrenziale, fatte salve le conseguenze sul rapporto di lavoro. Difettano i requisiti di tutela del prodotto editoriale come "banca dati" ai sensi della legge sul diritto d'autore nel caso di assoluta assenza di soluzioni originali nell'impianto redazionale, organizzativo, sistematico dei dati contenuti, ed in particolare a) in mancanza di caratteristiche creative nella suddivisione per archivi, b) in mancanza di elementi di differenziazione nella forma estetica (schermate, interfacce) e c) sussistendo il carattere meramente descrittivo e vincolato della nomenclatura utilizzata. L'illecito concorrenziale non è di per sé rivelatore della violazione dei diritti sulla proprietà intellettuale: pertanto la prova che le soluzioni tecniche adottate rivestono un grado di originalità tale (da intendersi, al minimo, come scelta tra più soluzioni tecniche possibili per la realizzazione del medesimo risultato funzionale) da differenziare il *software* come creazione intellettuale tutelabile ai sensi della legge sul diritto d'autore, non può implicitamente desumersi dalla accertata trasposizione di parti di un programma in altro prodotto concorrente. L'incarico di eseguire la manutenzione dei sistemi integrati di archivi normativi, nonché l'elaborazione di dati e tabelle sulla base delle esigenze editoriali della società committente, non comporta la facoltà per l'appaltatore di appropriarsi dei risultati della manutenzione o del prodotto informatico richiesto dalla committente; né la creazione o utilizzazione di programmi per elaboratore idonei ad assolvere alle specifiche indicate, ovvero la circostanza di aver acquistato a proprie spese tutto il materiale necessario (*software* e *hardware*) per la realizzazione della banca-dati, attribuisce all'appaltatore alcun diritto di libera utilizzazione del *data-base*».

(Trib. Roma 6.8.2001, *DA*, 2004, 374).

Siffatta estesa interpretazione dell'art. 2598, n. 3, c.c., non è unanime e non sono mancate autorevoli voci che hanno negato la capacità onnicomprensiva dell'art. 2598, n. 3 (Carnevali 1984, 251 ss.), in quanto incapace di comprendere tutti i comportamenti che non siano conformi alla correttezza professionale.

Tralasciando ogni più approfondita considerazione sulla portata più ampia, o più ristretta, delle fattispecie codicistiche quivi esaminate, ciò che si evince, da quanto brevemente illustrato, è, ancora una volta, l'insufficienza dei rimedi, di cui si è detto, a garantire una tutela "ad ampio raggio" all'autore del *software*. Le fattispecie di concorrenza sleale sono, infatti, applicabili soltanto nei confronti di soggetti imprenditori e, nel