

MARIO G. LOSANO

## OSSERVAZIONI SUL PROGETTO DI LEGGE GOVERNATIVO SULLA PROTEZIONE DEI PROGRAMMI

**SOMMARIO** 1. Punti di partenza e limiti di queste osservazioni. — 2. Le definizioni tecniche nelle norme sull'informatica. — 3. Le modifiche dei programmi e il diritto morale dell'autore-programmatore. — 4. Come si determina la responsabilità per i danni prodotti dalle modifiche di un programma? — 5. Due regole in contrasto. — 6. La registrazione dei programmi e la durata della loro protezione: alcuni dubbi. — 7. Le copie di sicurezza: una necessità che spesso non è virtù.

### 1. PUNTI DI PARTENZA E LIMITI DI QUESTE OSSERVAZIONI.

Sui problemi sollevati dalla tutela giuridica dei programmi (sia registrati su supporto magnetico, sia incorporati in *chips*) il giurista italiano ha finora avuto esprimersi in termini generali: l'assenza di una specifica legislazione nazionale sull'argomento lo obbliga ad analisi teoriche, a proposte di estensione analogica delle norme vigenti o ad esercizi di comparazione con ordinamenti stranieri, dove queste norme già esistono e danno i loro frutti giurisprudenziali.

La mia posizione su questi temi si può sintetizzare nell'accettazione della tutela offerta dal diritto d'autore in base soprattutto a tre ragioni: perché il diritto positivo italiano vieta l'uso del brevetto nel tutelare i programmi (e in ciò recepisce una norma sovranazionale); perché l'art. 99 della legge italiana sul diritto d'autore, in assenza di una specifica normativa, facilita l'estensione analogica di questa tutela anche ai programmi; perché la maggior parte degli Stati grandi produttori di *software* ha adottato la protezione del diritto d'autore, cosicché il discostarsene provocherebbe la non applicazione della clausola di reciprocità ai programmi italiani.

Questa clausola — sempre più presente in ogni legislazione che si occupi di informatica — obbliga ad una *Realpolitik* informatica che finisce per mettere i piccoli Stati sempre al rimorchio degli Stati Uniti. Giustamente le note di Vito Mangini su questa rivista si aprono proprio con un'indicazione a questa « USA-dependence ». Il Giappone — con ben altra potenza economica dell'Italia — ha cercato di mutar rotta e di introdurre una protezione dei programmi mediante brevetto, con il risultato di imbarcarsi in una guerra giuridico-commerciale da cui è uscito perdente.

D'altra parte, questa mia posizione pragmatica si riferisce unicamente alla fase di transizione che stiamo attraversando. Partendo dalle norme sul diritto d'autore, sarà necessario sviluppare una legislazione che tenga conto del fatto che non esiste un *software*, ma più tipi di *software*. E che queste realtà diverse esigono tutele differenziate, tra le quali non è da escludere un ritorno settoriale al brevetto<sup>1</sup>.

Nel 1986-87 la presentazione di quattro progetti di legge sulla protezione del *software* e del *firmware* consentì di dare al dibattito su questi problemi un taglio più concreto, permettendo di prendere posizione a favore o contro norme proposte finalmente in precisi termini linguistici. Anche se poi quella norma non verrà approvata o decadrà alla fine della legislatura, il dibattito da essa suscitato sarà utile quando il legislatore tornerà sull'argomento<sup>2</sup>. Perché una cosa è certa: uno Stato industrialmente avanzato non può fare a meno di queste norme e, se anche non le approva oggi, dovrà approvarle domani.

Poiché il dibattito aperto da *Il diritto dell'informazione e dell'informatica* ha già visto la partecipazione di illustri giuristi (Vittorio Afferni, Vincenzo Franceschelli, Vito Mangini, Giuseppe Sena), cercherò di analizzare soprattutto i problemi informatici connessi con alcune norme contenute nella proposta governativa. Lo farò senza pretese di completezza: in questo dibattito aperto non mancherà infatti la possibilità di ritornare su asserzioni non condivise dai lettori o su idee troppo sintetiche e quindi bisognose di sviluppo.

Per motivi di brevità, mi limiterò a discutere i problemi che sono stati sollevati dalla lettura dei primi 7 articoli della Sezione prima dello schema del disegno di legge presentato dalla Presidenza del Consiglio dei Ministri con il titolo « Norme per la tutela dei programmi per elaboratore e delle topografie di prodotti a semiconduttori ». Da queste osservazioni e dagli altri miei scritti ricordati in precedenza dovrebbe risultare chiara anche la mia posizione rispetto ai progetti Tedeschi e Fabbri, sempre sulla tutela del *software*.

Sulla tutela del *firmware* non mi sento invece in grado di valutare nel merito specifiche norme di legge, perché conosco troppo poco la tecnologia con cui vengono costruiti i *chips*. Ho a suo tempo apprezzato la legge americana in proposito<sup>3</sup>. Tuttavia la complessità tecnica

<sup>1</sup> Per il discorso più articolato che ha condotto a queste conclusioni rinvio al mio *Corso di informatica giuridica*, il cui vol. 2 è dedicato a *Il diritto privato dell'informatica* (Einaudi, Torino 1986). La protezione giuridica del software è trattata alle pp. 73-97, mentre la corrispondente bibliografia è alle pp. 241-261.

<sup>2</sup> Dei progetti Fabbri e Tedeschi mi ero

occupato in: *Due progetti di legge per la tutela del software*, in *Data Manager*, 1987, n. 63, p. 31; della situazione francese in un'intervista su *L'Elettronica: Protezione del software: in Francia è tutelato dal diritto d'autore* (maggio 1986, n. 31, pp. 7-9).

<sup>3</sup> *Copyright & Chips*, in *Data Manager*, 1985, n. 46, p. 9.

di questo settore me ne ha tenuto finora lontano: il disegno dettagliato — ma anche semplificato, pubblicato nel mio già citato volume a p. 92 — degli elementi strutturali di un *chip* consente di rendersi conto che le nozioni tecniche richieste sono tali, da esigere uno studio apposito, che io purtroppo non ho avuto modo di svolgere.

## 2. LE DEFINIZIONI TECNICHE NELLE NORME SULL'INFORMATICA.

Le definizioni con cui si apre l'art. 1 del progetto governativo possono essere discutibili, ma sono d'altra parte indispensabili per favorire la comprensione del testo legislativo anche da parte dei non giuristi. Infatti i testi di legge che si riferiscono a problemi tecnici — come appunto la programmazione o la produzione di semiconduttori (o *chips*) — sono destinati a divenire strumenti di lavoro e di consultazione anche per tecnici privi di una specifica preparazione giuridica.

Il lato negativo delle definizioni consiste nel vincolare il testo legislativo allo stato attuale della tecnologia, col rischio che le disposizioni si rivelino rapidamente obsolete, dato il susseguirsi di tecniche diverse. Il lato positivo mi pare consistere nella certezza del diritto che le definizioni introducono: grazie ad esse sappiamo di che cosa stiamo parlando. Questa certezza diviene un fondamentale fattore di armonizzazione legislativa quando un testo di legge si fonda su una proposta sovranazionale, come appunto il disegno di legge in esame. Se Stati diversi accettano le medesime definizioni, dovrebbe essere più semplice, in futuro, applicare le clausole di reciprocità.

Non mi sentirei dunque di condividere l'affermazione di Vincenzo Franceschelli, secondo cui « meglio comunque sarebbe stato lasciare alla magistratura il compito di dare, ai vari settori, flessibili definizioni » (917). Benché la preparazione informatica dei magistrati vada migliorando, non credo che la maggioranza di essi sia in grado di dare giudizi sicuri su specifici problemi di tecnologia informatica. Di conseguenza, affidando loro il compito di definire caso per caso i concetti informatici, si correrebbe il rischio non soltanto di ostacolare l'armonizzazione internazionale (o, almeno, europea), ma anche di frammentare la giurisprudenza nazionale.

Dunque, ben vengano le definizioni, specie se concordate a livello sovranazionale. Il parlamento deve però essere consapevole che questa tecnica legislativa esige una duplice cautela: da un lato, un buon lavoro di « manutenzione » delle leggi emanate, per evitare che il loro ritardo rispetto all'evoluzione tecnologica finisca per trasformarle in ostacoli allo sviluppo; dall'altro lato, un attento coordinamento tra definizioni tecniche contenute in leggi diverse.

A mio giudizio, il maggior pericolo si cela proprio nell'eterogeneità delle pur necessarie definizioni. Commentando i progetti Tedeschi e Fabbri su *Data Manager* (già citato alla nota 3) avevo svolto un'argomentazione che mi permetto di riportare: « Anche in Italia si sta

ormai generando un *corpus* di leggi sull'informatica, in cui è inevitabile usare la terminologia tecnica d'una disciplina poco nota ai giuristi e, per di più, in continua evoluzione. Per chiarezza di formulazione legislativa è spesso necessario definire i singoli termini tecnici propri dell'informatica. A volte queste definizioni sono esplicite, e quindi facilmente individuabili; altre volte sono invece implicite nella formulazione della norma, che ha magari recepito un parere tecnico. Le definizioni di entrambi i tipi possono costituire un pericolo, a lungo andare, se il redattore del testo legislativo non ha cura di coordinare le proprie definizioni con quelle già esistenti nel *corpus* della legislazione informatica italiana. Quest'ultima, nel corso del tempo, finirebbe per contenere definizioni giuridiche diverse (e forse incompatibili) dello stesso oggetto informatico. Oggi il diritto dell'informatica soffre dell'incertezza che deriva dalla mancanza di norme; in un futuro non lontano potrebbe soffrire dell'incertezza che nasce da norme discordanti. Tuttavia basterà un po' di attenzione alle tecniche legislative per evitare di cadere dalla brace nella padella ».

In realtà il problema delle tecniche legislative resta uno dei temi scottanti anche per l'informatica, poiché nessuna macchina e nessun programma può mettere riparo a leggi redatte male. Tuttavia alla tecnica legislativa sembrano più sensibili i funzionari regionali e centrali che i politici. Intanto, anche in Italia, va accumulandosi una normativa informatica non sempre coordinata che ha già raggiunto dimensioni di tutto rispetto<sup>4</sup>.

### 3. LE MODIFICHE DEI PROGRAMMI E IL DIRITTO MORALE DELL'AUTORE-PROGRAMMATORE.

L'art. 3, sulla titolarità dei diritti derivanti dalla stesura di un programma, distingue i diritti morali dell'autore da quelli economici derivanti dallo sfruttamento del programma. La distinzione non è esplicita, ma risulta dalla terminologia usata nei vari commi del futuro art. 7-*bis* della legge sul diritto d'autore. In generale, queste incertezze terminologiche sono anche spie di indecisioni concettuali. Cercherò di spiegarvi meglio.

Il diritto morale dell'autore — ad esempio, di un libro — gli consente di intervenire quando non ritenga corretta un'edizione antologica, una ristampa non riveduta e così via. Però il prodotto intellettuale classico assume quasi sempre la forma di qualcosa di fisico — libro, spartito, quadro, scultura ecc. — su cui un intervento esterno era difficile e richiede comunque un'ingerenza fisica sul prodotto dell'ingegno. Invece i programmi sono registrati con evanescenti ma-

<sup>4</sup> D.A. LIMONE, *Codice dell'informatica. Legislazione comunitaria, statale, regionale e circolari*, Maggioli, Rimini, 1985: vi

sono raccolte 1041 pp. di testi normativi riferentisi all'Italia.

gnetizzazioni su supporti che l'utente può modificare secondo le proprie esigenze. Anzi, deve poterli modificare: un programma non modificabile è infatti un programma poco vendibile. Uno dei pregi che i venditori di programmi attribuiscono ai loro prodotti è spesso quello della flessibilità, della possibilità di personalizzazione, della portabilità da un elaboratore all'altro. Tutti pregi che sia attuano, però, con un intervento sul programma; e non sempre si tratta di interventi puramente cosmetici.

Da queste considerazioni informatiche — qui ridotte in compresse — derivano i miei dubbi sull'ultimo comma dell'art. 3, dove si legge: « *Su autorizzazione del titolare del diritto originario, chiunque può apportare modifiche a programmi per elaboratori*, acquisendo il diritto di tutela per le sole parti modificate rispetto al programma originario ».

Alcuni esempi concreti possono forse illustrare meglio le mie perplessità sulla prima parte di questo comma (da me posta in corsivo).

a) Non è facile individuare « il titolare del diritto originario ». Ad esempio, il direttore di un centro di calcolo ha acquistato un programma e decide di apportarvi alcune modifiche e aggiunte per renderlo adatto alle elaborazioni che l'impresa si attende da quel centro. Egli deve quindi chiedere l'autorizzazione « al titolare del diritto originario »: ma chi è questo titolare? Se ci si riferisce soltanto ai *diritti economici*, è la *software house* neozelandese che ha sviluppato il programma, ovvero la società che distribuisce in esclusiva quel programma in Italia? (questa concatenazione di imprese che, a livello diverso, producono e distribuiscono il medesimo prodotto *software* è causa di frequenti impicci nell'informatica odierna. Chi acquista un programma da una società distributrice in italiana può scoprire che, nonostante la buona fede di entrambi, il programma è stato copiato e che il proprietario originario ha chiesto — magari dal Canada — il sequestro di tutte le copie contraffatte; quindi anche di quelle vendute in buona fede in Italia. Chi risarcisce il danno dell'acquirente italiano? Gli scritti sulla contrattualistica sono ricchi di informazioni su questo tema).

b) Supponendo risolto il precedente dubbio sul titolare del diritto originario, può l'autore del programma far valere i suoi *diritti morali*, negando l'autorizzazione a intervenire sul programma che egli ha sviluppato? A mio giudizio questo diritto morale esiste, dal momento che le norme sulla protezione del *software* sono collocate nella legge sul diritto d'autore e dal momento che queste proposte di modifica, specificamente introdotte per i programmi, parlano di « riconoscimento della paternità dell'opera » alla persona fisica autrice del programma. Ad esempio, se un programma per la gestione del personale aziendale viene venduto in Sudafrica, l'autore potrebbe rifiutare che il suo programma venisse modificato, introducendovi indicazioni sulla razza e sulla sindacalizzazione dei singoli addetti dell'impresa sudafricana. Oppure l'autore di un avanzato programma di calcolo potrebbe opporsi alla vendita di quel suo programma ad un paese co-

munista, per puri motivi sciovinistici o per timore che esso venga usato a fini bellici.

Ma come regolarsi in casi meno estremi, fondati non su grandi scontri etici, ma soltanto su antipatie aziendali o personali?

c) Rivolgerei una richiesta minima, di natura linguistica, agli esperti di proprietà intellettuale: non si può trovare un'espressione meno antropomorfica di « paternità dell'opera »? In campo informativo lavora un altro numero di donne programmatrici e gli scritti che parlano della loro paternità sui programmi finiscono per assumere un tono leggermente surreale.

d) Infine, quando la norma proposta dice che « chiunque può apportare modifiche ai programmi », apre la porta a interventi impreveduti, forse, da chi ne scrisse il testo. Finora abbiamo parlato dell'autore di un programma, ovvero delle imprese che lo producono e commercializzano. Sempre più spesso, però, la manutenzione di un programma è affidata ad una terza società, che non è né quella che lo produce, né quella che ne è l'utente finale. Ad esempio, può il mantentore — d'accordo con il « titolare del diritto originario » — effettuare interventi sul programma al solo fine di renderne più agevole la manutenzione? L'assenso dell'utente finale è irrilevante ai fini di queste modifiche? Se sì, dal momento che la manutenzione diviene più agevole, l'utente finale ha il diritto di chiedere una revisione al ribasso del contratto di manutenzione? Se dalla modifica del programma derivano danni a funzioni che non si credevano intaccate, con conseguente detrimento dell'utente finale, chi è tenuto al risarcimento dei danni?

Forse si eviterebbero questi e altri problemi se, invece di usare l'espressione « chiunque può apportare modifiche », si dicesse semplicemente « chi apporta modifiche ai programmi »: in questo modo non si attribuirebbe alcuna autorizzazione a modificare i programmi, ma si direbbe semplicemente che il modificatore riceve la medesima tutela dovuta all'autore, però, limitatamente alla sola parte modificata. Tutela che implica diritti e doveri, pretese e responsabilità.

#### 4. COME SI DETERMINA LA RESPONSABILITÀ PER I DANNI PRODOTTI DALLE MODIFICHE DI UN PROGRAMMA?

È già stato richiamato l'ultimo comma dell'art. 7, secondo cui « su autorizzazione del titolare del diritto originario, chiunque può apportare modifiche a programmi per elaboratori, *acquisendo il diritto di tutela per le sole parti modificate rispetto al programma originario* ». Dopo le osservazioni svolte nel paragrafo precedente, vorrei soffermarmi ora sulla seconda parte della norma, anche qui da me posta in corsivo.

Questo comma presenta una visione delle aggiunte o modificazioni di un programma ancora troppo legata alla forma materiale che in generale assume, tradizionalmente, il prodotto dell'ingegno. Un conto è inserire un nuovo capitolo in un libro o un quarto movimento in

una composizione musicale, altro è aggiungere un segmento ad un programma già esistente. La programmazione strutturata e le tecniche più moderne di programmazione mirano a eliminare o, almeno, a ridurre al minimo le ripercussioni indesiderate di una parte del programma sulle restanti, cosicché il programma si presenta come un tutto coerente, le cui parti sono allacciate da fili ramificati per ogni dove. Tuttavia questi programmi sono prodotti altamente perfezionati, ma non perfetti. I programmi di grandi dimensioni vengono elaborati da gruppi che, nella fase finale, assemblano i vari lavori parziali; i pochi che conoscono il disegno unitario non hanno potuto approfondire i singoli segmenti; chi conosce a fondo il proprio segmento, ha invece un'idea vaga del disegno generale e non conosce i segmenti sviluppati dagli altri gruppi.

Insomma, è sempre più difficile trovare un unico individuo che sappia tutto di un grosso programma. In pratica, avviene che il programma finito venga sondato come una scatola nera: si prova a fargli fare certe operazioni e si esamina come reagisce. Dalle reazioni anomale si risale ai difetti di programmazione. Tuttavia è possibile che qualche anomalia sfugga a questi controlli.

Può essere quindi difficile stabilire con esattezza qual è la parte nuova, aggiunta al programma preesistente, e su cui il nuovo autore ha diritti e doveri specifici; ma può essere ancora più difficile stabilire se certe anomalie di funzionamento del programma vadano ricondotte univocamente alle modifiche successive, oppure non fossero già implicite — vizio occulto? — nel programma fornito originariamente.

La norma in esame è formulata in modo ottimistico, perché prospetta il diritto di tutela delle parti aggiunte, ma tace sulle corrispondenti responsabilità del loro autore. Nella realtà questi ultimi casi non sono rari. Alcuni esempi possono illustrare le questioni giuridiche suscitate da questa norma:

— nonostante l'autorizzazione preventiva, in caso di intoppi nel programma successivi all'intervento del nuovo autore sorgeranno di certo controversie tra il produttore originario e l'autore interveniente, nel tentativo di scaricare l'uno sull'altro la responsabilità del danno subito dall'impresa;

— se ogni autore ha un suo diritto morale sulla parte di programma che gli è (difficoltosamente) ascrivibile, possono verificarsi ostacoli all'intervento del produttore originario dell'intero programma sul segmento aggiunto. Sul piano della singola azienda, infatti, è spesso sostenibile la tesi che l'aggiunta — limitata ma personalizzante — riveste una grande importanza per la gestione automatizzata di alcune essenziali funzioni dell'impresa;

— poiché il diritto morale permane anche quando l'autore di un segmento addizionale di programma ha lasciato (magari in malo modo) l'impresa, questa norma rischia di costituire uno strumento di disturbo o di ricatto non indifferente nei rapporti tra impresa e individuo, tanto al momento dell'interruzione del rapporto, quanto in mo-

menti futuri. Un programmatore dall'*esprit mal tordu* potrebbe inserire nei programmi non una « bomba a tempo », ormai ben nota tra i reati informatici, ma il suo equivalente giuridico: realizzerebbe in pratica le medesime finalità di disturbo e di riscatto dell'impresa, senza commettere alcun reato informatico, ma anzi avvalendosi delle prerogative che la legge — se entrerà in vigore in questa forma — gli attribuirebbe.

## 5. DUE REGOLE IN CONTRASTO.

Le considerazioni fin qui svolte non tengono conto di una contraddizione fra l'art. 3 e l'art. 7 dello schema di disegno di legge governativo. All'art. 7 esso propone infatti di modificare la legge sul diritto d'autore introducendo un art. 64-ter che autorizza il legittimo detentore di un programma « a fare tutte le modifiche necessarie a rendere il programma per elaboratore più idoneo alle proprie necessità, avvalendosi eventualmente dell'opera di terzi, previa *comunicazione* al titolare dei diritti esclusivi d'autore ».

Se intendo correttamente il significato di questa norma, qui si dice una cosa diversa da quella esposta all'ultimo capoverso dall'art. 3 del medesimo schema di legge, dove si richiedeva « l'*autorizzazione* del titolare del diritto originario ». Secondo l'art. 7, affinché l'intervento sia legittimo basta che l'impresa ritenga opportuno intervenire sul programma e che ne informi il titolare originario. In altre parole, non c'è limite all'intervento sui programmi, poiché di fatto nessuno meglio dell'impresa in questione può valutare se l'intervento è idoneo o no rispetto alle proprie finalità.

Questa norma appare ispirata ad un pragmatismo più vicino alla realtà dell'informatica dei giorni nostri e meno legato a formalismi giuridici di difficile, e spesso impossibile, verifica. Per questa ragione la riterrei preferibile all'art. 3, indubbiamente più garantista ma anche — come spero di aver illustrato — di difficile applicazione.

## 6. LA REGISTRAZIONE DEI PROGRAMMI E LA DURATA DELLA LORO PROTEZIONE: ALCUNI DUBBI.

Un analogo realismo si ritrova nell'art. 4, dove la durata dei diritti viene fatta dipendere o da una prova di fatto o dalla registrazione presso la SIAE, regolata per esteso dall'art. 5.

Condivido e plaudo a questa diffidenza verso gli archivi pubblici: non perché ritenga inutili gli archivi, ma perché la gestione degli archivi esistenti da parte della pubblica amministrazione finisce per trasformarli non in raccolte, ma in cimiteri di dati. Neppure l'informatica è riuscita a mutare questa situazione. Gli archivi cartacei o sono restati cartacei (come il Catasto), o non sono riusciti a diventare operativi (come il progetto Atena per il fisco o il catalogo unico delle



biblioteche italiane), o — se operativi — sono stati costruiti con scarsa professionalità ed hanno talora prodotto gravi disfunzioni: il centro della motorizzazione civile, ad esempio, non aveva strutture di sicurezza che gli permettessero di superare i danni derivati da un'attentato. Bene dunque che la prova dell'esistenza del programma non vada affidata al solo registro presso la SIAE.

Un dubbio è spontaneo: poiché il deposito presso la SIAE è « facoltativo e oneroso » (art. 6) e poiché esso fa fede « sino a prova contraria » (art. 5) ed è sostituibile con una prova fattuale (art. 4), dovrebbe esservi ben poco interesse, da parte di un individuo o di un'impresa, ad accettare questa procedura.

Sul piano della tecnica legislativa, poi, sembra eccessivo istituire una sovrastruttura burocratica sancendone contemporaneamente l'irrelevanza di fatto.

Senz'altro accettabile è la riserva che Giuseppe Sena (p. 941) avanza nei riguardi dell'art. 4, che fa partire dalla data di commercializzazione la prova fattuale dell'esistenza di un programma. L'argomento di Sena è giuridico: poiché anche oggi, in assenza di una specifica norma, dottrina e giurisprudenza classificano il programma tra le opere dell'ingegno, la data della sua esistenza giuridicamente rilevante deve essere quella della creazione, e non quella della commercializzazione.

Se ve ne fosse bisogno, questa osservazione può essere confortata da una constatazione empirica: non tutto il *software* già esistente viene immediatamente commercializzato. Un programma può essere a lungo usato nell'ambito della ricerca e sviluppo di un'impresa prima di essere diffuso (anche fra le imprese concorrenti). Un programma può nascere da una ricerca finanziata dal Consiglio Nazionale delle Ricerche, che prevede espressamente l'obbligo di mettere gratuitamente a disposizione di tutti il programma: il contrario quindi della commercializzazione. Un programma può nascere in una facoltà o in un centro di calcolo universitario ed essere oggetto anzitutto di una pubblicazione scientifica o elettronica<sup>5</sup>; da essa un'impresa può cogliere l'idea e renderla commerciale: a chi dei due spetterebbe allora la tutela accordata dalla proposta di legge?

All'art. 4, forse, il termine « commercializzare » andrebbe sostituito con « rendere pubblico » o qualcosa di simile, che è termine di *genus* rispetto alla *species* precedente: commercializzare è infatti un caso particolare in cui si rende pubblico qualcosa a fine di lucro. Proprio questo fine manca in molti casi, in cui però il prodotto esiste ed è meritevole di tutela giuridica.

<sup>5</sup> Gli istituti scientifici connessi in rete hanno l'abitudine di mettere alcuni dei programmi realizzati a disposizione dei colleghi. Si tratta di una forma di pubblicazione diver-

sa da quelle tradizionali: con essa l'autore del programma rinuncia allo sfruttamento economico, ma non ai diritti morali sulla sua opera.

## 7. LE COPIE DI SICUREZZA: UNA NECESSITÀ CHE SPESSO NON È VIRTÙ.

Il progetto di legge governativo risolve forse troppo rapidamente il problema delle copie di sicurezza. L'art. 7 prevede un futuro 64-bis, che in due commi autorizza l'utente a riprodurre il programma. Il punto a) si riferisce alla copia di sicurezza; il punto b) a qualcosa che non capisco bene.

a) È lecita la riproduzione di un programma « effettuata al solo scopo di disporre di una copia di riserva ». Non è chiaro se si intende autorizzare un'unica copia di riserva: una sola potrebbe essere troppo poco, oppure troppo. Si tocca qui con mano come la natura dei programmi sia troppo eterogenea per essere trattata in modo unitario. Per ragioni di brevità mi limito a rinviare alle pagine in cui ho trattato le varie tecniche di protezione logica e fisica dei programmi<sup>6</sup>.

Vi sono programmi costosissimi, in cui il problema dell'utente viene risolto con un contatto diretto con la casa produttrice. All'altro estremo, vi sono programmi così diffusi e così a buon mercato, che il produttore dà per scontato che essi vengano largamente copiati. Per certe categorie di programmi, anzi, si tende ormai ad ammettere che li si riproduca senza ostacoli fisici, logici e giuridici<sup>7</sup>. Tenendo conto di questa realtà, la norma proposta appare insufficiente, anche se è immaginabile che ad essa i contratti derogheranno in vario modo, per salvaguardare le particolarità del singolo programma.

Se la norma viene invece interpretata come inderogabile, restano aperti due interrogativi: che ne è di quei programmi che non prevedono la copia di sicurezza, ma l'intervento della casa produttrice? Qual è la responsabilità dell'utente che venga scoperto ad usare la copia di sicurezza su un'altra sua macchina?

In questo contesto le *software houses* non potrebbero accettare la richiesta di Vincenzo Franceschelli, quando scrive: « Sono decisamente contrario ad ogni limitazione riproduttiva dei programmi effettuata dall'utilizzatore a fine privato o non commerciale. Si vieti — e giustamente — ogni riproduzione a fine di lucro — produzione, commercializzazione, cessione e vendita. Ma ogni intervento diretto al fine privato — sia per disporre di una o più copie di riserva, sia per puro piacere, sia per qualsiasi altro motivo — va decisamente respinto » (pag. 921).

Se si esce dal campo dei programmi più diffusi e più elementari destinati ai *personal computers*, questa liberalizzazione indiscriminata danneggerebbe le *software houses*. Nella gestione della grande impresa, nell'ambito dell'elaborazione grafica o nel settore della ricerca

<sup>6</sup> LOSANO, *Diritto privato dell'informatica*, cit., pp. 72-78.

<sup>7</sup> La rinuncia alla protezione del software, in *Data Manager*, 1987, n. 66, p. 33.

e sviluppo, vi sono programmi assai costosi, ovvero con noleggi dell'ordine di centinaia di milioni di lire al mese. Il loro utente naturale è la grande impresa multinazionale. Un'eccessiva liberalizzazione porterebbe l'impresa ad acquisire un esemplare del programma ed a distribuirne poi più copie tra le proprie filiali (sparse in tutto il mondo) o magari anche alle società controllate. Una circolazione altrettanto vorticosa e gratuita potrebbe verificarsi nell'ambito dell'apparato statale.

Insomma, al principio della liberalizzazione sarebbe forse opportuno contrapporre il principio di un controllo flessibile sulla circolazione delle copie. In quest'ultimo non si dovrebbe poi perdere di vista la difficoltà di verificare in concreto quando c'è violazione degli obblighi assunti. L'accertamento è di fatto impossibile nei programmi per i *personal computers*, ma diviene sempre meno difficoltoso man mano che si passa ai grandi centri, ovvero ai prodotti specializzati in cui la stessa impresa fornisce, ad esempio, la macchina e i programmi. In questi ultimi casi gli interventi di manutenzione consentono un controllo penetrante che rende efficaci le norme sulla limitazione della copiatura di un programma.

b) Il capoverso prevede poi, alla lett. b), che sia legittima la riproduzione del programma quando essa « costituisca una fase necessaria all'utilizzazione autorizzata del programma per elaboratore ». Confesso che la norma mi sembra o superflua o incomprensibile.

È superflua se si riferisce alle copie di sicurezza (del resto ricordate poche righe prima, nel medesimo comma), ovvero al fatto che certe parti del programma, nella sua esecuzione, vengano copiate anche più volte in certe parti di memoria.

È incomprensibile se si riferisce a qualcosa di diverso, poiché né io, né alcuni colleghi e tecnici interpellati, riusciamo ad immaginare una situazione diversa dalle due precedenti. Non avendo potuto leggere le relazioni che hanno preparato e accompagnato il testo di legge, non esprimo tuttavia un giudizio definitivo: può darsi che il capoverso esprima in modo linguisticamente inadeguato una realtà che io ignoro. In ogni caso ne sarebbe consigliabile la revisione almeno formale, data l'oscurità di questa stesura.