

ALMA MATER STUDIORUM · UNIVERSITÀ DI
BOLOGNA

SCUOLA DI SCIENZE
Corso di Laurea in Informatica

**Gestione e diffusione
di un codice Open Source**

Relatore:
Chiar.mo Prof.
Giulio Casciola

Presentata da:
Simona Giordani

Sessione II
Anno Accademico 2012-2013

Introduzione

Negli ultimi anni la diffusione del software libero o open source ha riscosso un grande successo, tanto che ormai molti programmi comunemente utilizzati rientrano in questa categoria, o comunque solitamente è possibile trovare un'alternativa open alla maggior parte dei programmi proprietari. Infatti al giorno d'oggi questo tipo di software è talmente diffuso e popolare che è possibile trovarlo in campi diversissimi: per esempio nel mondo degli smartphone Android è il sistema operativo più diffuso, mentre in quello dei web browser Firefox è il secondo più diffuso. In questo elaborato si intende cercare di fornire il maggior numero di informazioni utili per chiunque volesse rilasciare pubblicamente il proprio programma, in modo da permettere agli utilizzatori di adattarlo alle proprie esigenze o a quelle altrui, o anche solo per chi volesse approfondire la propria conoscenza su questo tipo di licenze e sulla terminologia che comunemente le accompagna. Uno dei principali vantaggi nello sviluppo di questo tipo di software è infatti quello di avere a disposizione un numero enorme di possibili collaboratori, che potrebbero aiutare nel miglioramento del programma con contributi che possono comprendere dalla segnalazione di bug o altri problemi, alla correzione dei problemi già conosciuti, al miglioramento e l'aggiunta di nuove funzionalità.

Nel primo capitolo verranno introdotti i concetti di Open Source e Software Libero e verranno esposte alcune nozioni che possono essere utili per comprendere il loro funzionamento e le idee alla loro base.

Il secondo capitolo spiegherà brevemente la licenza GPL versione 3 e altre licenze simili, e chiarirà le loro differenze.

Infine il capitolo finale mostrerà anche con un esempio pratico come applicare la licenza al proprio programma, e fornirà alcune informazioni per come diffonderlo nel web.

Indice

Introduzione	i
1 Introduzione al mondo delle licenze open source e quelle per il software libero.	1
1.1 Licenze	1
1.1.1 Breve storia delle licenze sul software	2
1.2 Cos'è l'Open Source	3
1.2.1 Licenze per il software libero e/o open source	3
1.3 Compatibilità fra licenze	8
1.4 Terminologia e concetti importanti	9
1.4.1 Codice Sorgente	9
1.4.2 Codice oggetto	10
1.4.3 Copyright	10
1.4.4 Copyleft	11
2 Scelta della licenza	15
2.1 GPLv3 riassunta punto per punto	15
2.2 Breve descrizione AGPLv3 (GNU Affero General Public License Versione 3)	23
2.3 Breve descrizione LGPLv3 (GNU Lesser General Public License versione 3)	24
2.4 Principali differenze tra GPLv2 e GPLv3	24
2.4.1 Tivoizzazione	27

3	Applicazione della Licenza e distribuzione del programma	29
3.1	XCModel un sistema per modellare curve e superfici NURBS	29
3.1.1	Storia di XCModel	37
3.1.2	Gestione del progetto e ostacoli	38
3.2	Preparare un pacchetto per la distribuzione con licenza GPLv3	41
3.2.1	Includere codice preso da un altro progetto senza averlo modificato	43
3.2.2	Includere codice preso da un altro progetto dopo averlo modificato	44
3.3	Licenziare XCModel sotto GPLv3	45
3.4	Diffondere il proprio programma Open Source	49
3.5	Il dual licensing (o multi licensing)	52
3.5.1	Dual licensing per la compatibilità con più licenze	52
3.5.2	Dual licensing come metodo per lo sfruttamento com- merciale	53
3.6	Modelli di guadagno	54
3.7	Come comportarsi se qualcuno infrange la licenza	57
	Conclusioni	59
	Bibliografia	61

Elenco delle figure

1.1	Esempi di licenze che rientrano in almeno una delle due categorie	4
1.2	Logo della Free Software Foundation	4
1.3	Logo Open Source Initiative	6
1.4	Tabella delle compatibilità tra GPLv2 e GPLv3	9
1.5	Relazione tra codice sorgente e codice oggetto	10
1.6	Simbolo Copyright	10
1.7	Simbolo Copyleft	11
1.8	“Viralità” della licenza	13
1.9	“Non viralità” della licenza	14
2.1	Logo della licenza GPL versione 3	15
2.2	Logo della licenza AGPL versione 3	23
2.3	Logo della licenza LGPL versione 3	24
3.1	Schermata XCMoel	30
3.2	Schermata XCCurv	30
3.3	Schermata XCSurf	31
3.4	Schermata XCBool	32
3.5	Schermata XCRayt	33
3.6	Schermata XCView	34
3.7	Struttura XCMoel	35
3.8	Sito ufficiale di XCMoel	39
3.9	Sezione 3D Modeling Projects	40
3.10	Sezione 3D NURBS Models	41

3.11	Contenuto cartella XCMoel	46
3.12	Contenuto sotto-cartella sources	47
3.13	Notifica inserita nei file sorgente di XCMoel	49
3.14	Scissione tra software open source e prodotto commerciale . .	55
3.15	Prodotto fornito gratuitamente ma vendita dei servizi	56

Capitolo 1

Introduzione al mondo delle licenze open source e quelle per il software libero.

1.1 Licenze

In ambito informatico una licenza di un software è considerata un contratto con cui chi possiede i diritti su quel software indica quali sono le regole che l'utilizzatore deve rispettare per avere il diritto di utilizzarlo. Solitamente queste regole comprendono il numero di copie del programma che si possono utilizzare, eventuali limitazioni di responsabilità, limitazioni sull'utilizzo e sulla sua distribuzione. Solitamente le licenze possono essere suddivise in due macro categorie, cioè licenze proprietarie (Closed Source) che qui non tratteremo e licenze per il software libero e/o open source. Le licenze proprietarie tendono ad essere molto più restrittive per l'utilizzatore del software rispetto a quelle open source, in quanto solitamente non permettono la modifica e la redistribuzione del software e non rendono disponibile il codice sorgente.

1.1.1 Breve storia delle licenze sul software

Fino alla fine degli anni '70 inizio degli anni '80 l'utilizzo delle licenze sul software non era molto diffuso, in quanto le leggi sul copyright dell'epoca non chiarivano se queste si applicassero anche al software, in più i programmi tendevano a essere utilizzabili solo su macchine compatibili e la loro copia era molto difficoltosa. Con l'avvento dei sistemi operativi, che permettevano di utilizzare lo stesso codice su macchine diverse, del progressivo ampliarsi del mercato potenziale, dell'introduzione di supporti per la distribuzione facilmente divulgabili e che facilitavano quindi la copia del software, si era però creata la necessità di adeguarsi alla nuova situazione e quindi molte aziende iniziarono ad utilizzare licenze restrittive sui programmi da loro distribuiti e a non distribuire il loro codice sorgente. Questa nuova tendenza non piacque però a molti programmatori, e nel 1985 portò alla fondazione della Free Software Foundation (FSF) da parte di Richard Stallman. Questa organizzazione senza scopo di lucro, è nata con l'obiettivo di diffondere il software libero concentrandosi sugli aspetti politici e legali e nel 1989 rilasciò la prima versione della GNU General Public License.

Con l'inizio degli anni '90 e la nascita dei web browser e il diffondersi di Internet, vi fu un'ulteriore spinta per la distribuzione del software libero, che portò tra le altre cose anche alla nascita di Linux, creato da Linus Torvalds e migliorato da molti programmatori provenienti da varie parti del mondo.

Nonostante tutto, nel mondo commerciale il software libero era ancora visto con diffidenza, sia per il fraintendimento comune sul suo nome (free in inglese vuole dire sia libero che gratis, ma il software libero di solito può essere venduto a meno che la licenza specifica non lo vieti), sia a causa del sospetto che le idee di Stallman suscitavano. Per contrastare questo problema quindi, verso la fine degli anni '90, in occasione del rilascio del codice sorgente del web browser Netscape , si decise di sfruttare l'occasione per sostenere ed educare il pubblico riguardo l'utilizzo e lo sviluppo di tipo open anche dal punto di vista commerciale. Per distanziare questo approccio da quello più ideologico del software libero si decise quindi di usare il termine "open source" per

definirlo, in modo da porre l'attenzione sulla disponibilità dei sorgenti. Nel 1998 venne quindi fondata l'Open Source Initiative (OSI), con l'obbiettivo di diffondere e proteggere la definizione di open source.

1.2 Cos'è l'Open Source

Con il termine open source ci si riferisce a un modello di sviluppo e di produzione basato sul libero accesso e libera distribuzione dei progetti di un determinato prodotto. Partendo da quei progetti chiunque può decidere di creare una versione modificata del prodotto e ridistribuire il risultato al pubblico. L'ambito in cui questa metodologia di condivisione è più diffusa è quello informatico in cui, grazie anche a Internet, un progetto può essere utilizzato o modificato da più persone sparse per il mondo, ma è comunque presente anche in ambito elettronico, medico, scientifico e in molti altri campi di applicazione.

1.2.1 Licenze per il software libero e/o open source

Un errore che viene commesso comunemente è quello di pensare che “software libero” e “software open source” abbiano lo stesso significato. In realtà anche se hanno caratteristiche simili, i loro obiettivi e la filosofia alla loro base sono molto differenti, e può capitare che una licenza che viene considerata conforme ai criteri di una non venga considerata conforme a quelli dell'altra.

Il primo tipo ad essere codificato in una serie di regole furono le licenze per il software libero, che nacquero negli anni '80, per contrastare la sempre crescente diffusione del software closed source che con le sue restrizioni impediva la diffusione e la modifica dei software, mentre verso la fine degli anni '90 venne coniato il termine “open source”, per indicare un tipo di software libero che potesse essere appetibile anche dal punto di vista commerciale. Mentre molti confondono il concetto di “free software” con quello di gratuito, in realtà questo non è un suo prerequisito, dato che non è vietato lo sfruttamento commerciale del software libero.

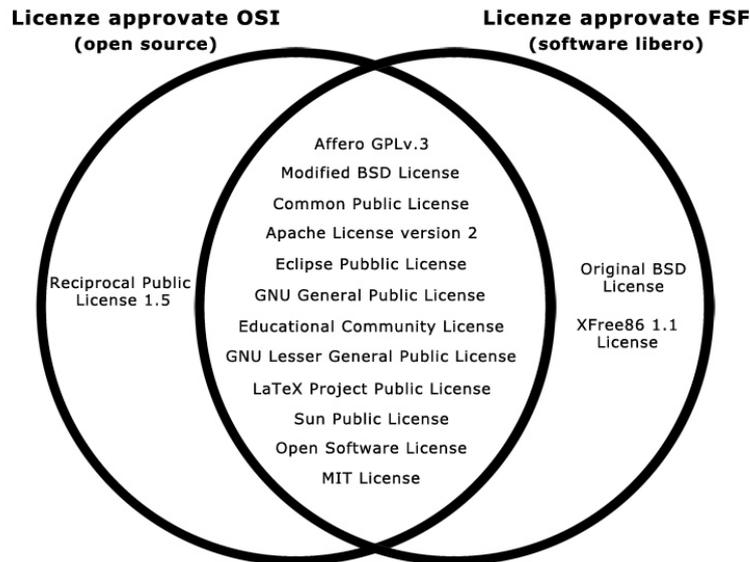


Figura 1.1: Esempi di licenze che rientrano in almeno una delle due categorie

Il concetto alla base del software libero è quello di Libertà intesa come libertà imprescindibile per l'utente di utilizzare, copiare, modificare e distribuire il software, mentre il software open source tende a essere più una metodologia di sviluppo e dare la priorità allo sviluppo e alla commerciabilità (intesa come diffusione e utilizzo anche da parte delle aziende e degli utenti), che viene favorita dalla diffusione del codice sorgente e dallo sviluppo cooperativo di più sviluppatori. In pratica la differenza principale tra software libero e quello open source è che mentre il primo è basato su un ideale, il secondo segue principi più pragmatici. Queste differenze sono visibili anche confrontando i principi su cui si basano le due filosofie.



Figura 1.2: Logo della Free Software Foundation

Per esempio dalla "Free Software Definition" un programma è considerato

software libero se rispetta tutte le seguenti quattro libertà:

- **Libertà di usare il programma, per qualsiasi scopo.** In questo caso si intende che ogni persona e/o organizzazione può usare il programma in qualsiasi tipo di sistema per qualsiasi fine, senza dover chiedere il permesso allo sviluppatore o altri specifici soggetti.
- **Libertà di studiare come funziona il programma, e cambiarlo in modo che soddisfi le proprie esigenze. L'accesso al codice sorgente ne è un prerequisito.** Questa libertà comprende anche il diritto dell'utente di modificare e usare la propria versione del programma. Un chiarimento che viene aggiunto nella "Free Software Definition" è che se una licenza non permette di unire moduli che hanno una licenza adeguata, a meno che per esempio non se ne posseda il copyright, allora non è considerata una licenza libera.
- **Libertà di ridistribuire copie in modo da aiutare il prossimo.** Includendo il diritto di ridistribuire gli eventuali eseguibili (se producibili), oltre ovviamente al codice sorgente, quindi una licenza non può vietare la distribuzione di una delle due forme.
- **Libertà di distribuire copie della propria versione modificata ad altri. In questo modo si può dare la possibilità di beneficiare del proprio lavoro a tutta la comunità. L'accesso al codice sorgente ne è un prerequisito.** Concede all'utente la possibilità di rilasciare la propria versione modificata del software almeno in forma libera, qualora lo volesse.

Un punto fondamentale che risulta da queste quattro libertà (tanto da essere ripetuto in due di esse) è quello dell'obbligo di fornire l'accesso al codice sorgente del software (sorgenti offuscati¹ non vengono considerati validi). Queste libertà devono essere irrevocabili e a meno che non si infranga qualche

¹Codice sorgente che è stato appositamente reso difficile da comprendere a un essere umano, in modo che esso non possa modificarlo.

regola, non è possibile aggiungere restrizioni o revocare la licenza da parte dello sviluppatore.

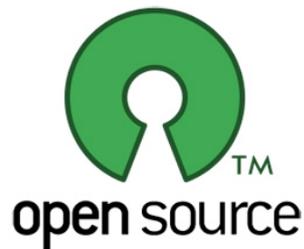


Figura 1.3: Logo Open Source Initiative

Le regole che un software “open source” deve seguire sono specificate dalla “Open Source Initiative” e sono definite dai seguenti principi:

1. **Libertà di redistribuzione:** La licenza non può limitare nessuna delle parti dal vendere o distribuire il software come componente di una distribuzione di software aggregati che contiene programmi da fonti diverse. La licenza non deve richiedere il pagamento di royalty o altre imposte sulla vendita.
2. **Codice Sorgente:** Il programma deve includere il codice sorgente, e deve permettere la distribuzione sia in forma di codice sorgente che nella forma compilata. In caso una versione del prodotto sia distribuita senza il codice sorgente, allora devono esserci dei mezzi ben reclamizzati per ottenere il codice sorgente, preferibilmente per non più di un costo ragionevole di riproduzione, preferibilmente tramite il download da internet senza costi aggiuntivi. Il codice sorgente deve essere la forma principale da cui un programmatore possa modificare il programma. Del codice sorgente deliberatamente offuscato non è consentito. Forme intermedie come l’output di un preprocessore o di un traduttore non sono consentite.

3. **Opere derivate:** La licenza deve consentire l'apporto di modifiche e la creazione di lavori derivati, e deve consentire la loro distribuzione entro gli stessi termini della licenza del software originale.
4. **Integrità del codice sorgente dell'autore:** La licenza può porre restrizioni sulla distribuzione di forme modificate del codice sorgente solo se la licenza permette la distribuzione di "file patch", con lo scopo di modificare il programma durante il build time, insieme al codice sorgente. La licenza deve permettere esplicitamente la distribuzione del software realizzato a partire da codice sorgente modificato. La licenza può richiedere che lavori derivati abbiano un nome o un numero di versione diverso dall'originale.
5. **Nessuna discriminazione contro persone o gruppi:** La licenza non deve discriminare nessuna persona o gruppo.
6. **Nessuna discriminazione contro campi di utilizzo:** La licenza non può impedire l'utilizzo del programma in nessun campo di applicazione. Per esempio, non può impedire l'utilizzo del programma in ambito commerciale, o l'utilizzo per la ricerca genetica.
7. **Distribuibilità della licenza:** I diritti assegnati al programma devono applicarsi a tutti coloro a cui il programma viene ridistribuito senza la necessità che essi applichino una licenza addizionale.
8. **La licenza non deve essere specifica per un prodotto:** I diritti assegnati al programma non devono dipendere dal fatto che esso faccia parte di una specifica distribuzione di software. Se il programma è estratto da quella distribuzione e usato o distribuito all'interno dei termini della sua licenza, tutte le parti a cui il programma è ridistribuito devono avere i medesimi diritti di coloro ai quali erano stati concessi insieme alla distribuzione originale.
9. **La licenza non può limitare altri software:** La licenza non può porre restrizioni su altri software che sono distribuiti insieme al software

licenziato. Per esempio, la licenza non può porre l'obbligo che tutti gli altri programmi distribuiti nello stesso supporto debbano essere open source.

10. **La licenza deve essere tecnologicamente neutrale:** Nessuna clausola della licenza può essere basata su singole tecnologie o stili di interfacce.

1.3 Compatibilità fra licenze

Una licenza si dice compatibile con un'altra licenza se il codice (o parte di esso) coperto da una licenza può essere usato all'interno di un programma coperto da una licenza diversa, senza infrangere i termini di nessuna delle due. Il fatto che due licenze siano entrambe considerate libere (o open source) non vuol dire che esse siano anche automaticamente compatibili tra loro, infatti ad esempio la licenza BSD Originale pur essendo considerata una licenza libera non è considerata compatibile con le licenze GPL a causa della seguente condizione che impone di includere una specifica frase di attribuzione in ogni materiale pubblicitario riguardante il software coperto da essa:

3. All advertising materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by the University of California, Berkeley and its contributors.

Un altro esempio ancora più emblematico di incompatibilità tra licenze è quello tra la licenza GPLv2 e la GPLv3, in quanto entrambe hanno una clausola di propagazione della licenza che impone di rilasciare il software che fa uso di codice rilasciato sotto licenza GPL con la stessa licenza:

GPLv2 2.b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

GPLv3 5.c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

		Licenza applicata al codice che si vuole rilasciare		
		GPLv2 only	GPLv2 or later	GPLv3 or later
Licenza applicata al codice che si vuole utilizzare per il proprio progetto	GPLv2 only	OK	OK	NO
	GPLv2 or later	OK	OK	OK
	GPLv3	NO	OK: Convey project under GPLv3	OK

Figura 1.4: Tabella delle compatibilità tra GPLv2 e GPLv3

1.4 Terminologia e concetti importanti

1.4.1 Codice Sorgente

Con il termine codice sorgente si intende un insieme di istruzioni che siano scritte in un linguaggio comprensibile sia da parte del computer, sia da parte di un essere umano che conosca quel linguaggio, e che se opportunamente elaborate (tramite compilazione o interpretazione), possano dare origine a un programma. Questo è solitamente il formato di distribuzione di software da parte di coloro che ne vogliono permettere lo studio o l'alterazione, in quanto ne è la versione più comprensibile e facilmente modificabile.

1.4.2 Codice oggetto

Questo termine si riferisce al prodotto della compilazione di un file ed è solitamente in linguaggio macchina e quindi non facilmente comprensibile da parte di un essere umano. Esso può essere passato a un linker che provvederà a integrare i vari moduli che compongono il programma per dare origine all'eseguibile, anche se a volte la compilazione e il linking vengono fatti entrambi dal compilatore.

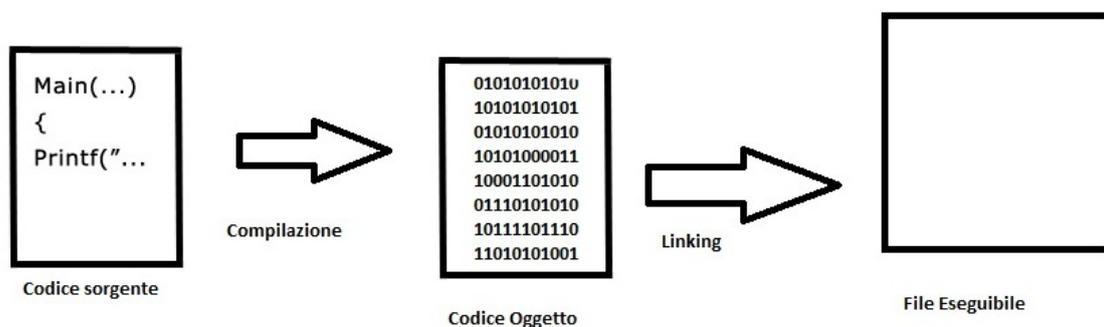


Figura 1.5: Relazione tra codice sorgente e codice oggetto

1.4.3 Copyright



Figura 1.6: Simbolo Copyright

Il concetto di Copyright è utilizzato in ambito legale per indicare il diritto di essere accreditati per un'opera da parte di colui che ne detiene i diritti, e quali diritti sono concessi a coloro che usufruiscono di quest'opera, ed è quindi considerato una forma di proprietà intellettuale. Può essere applicato

a varie tipologie di lavori, siano essi in ambito intellettuale, artistico, creativo o altro, ma non può essere applicato su idee o informazioni a sé stanti, in quanto può essere utilizzato solo sulla specifica forma in cui sono applicati. L'utilizzo del copyright è riconosciuto principalmente in ambito territoriale, quindi i termini specifici possono variare da nazione a nazione, ma grazie a accordi internazionali (tra cui i principali sono stati ratificati durante la “Convenzione di Berna per la protezione delle opere letterarie e artistiche²” e la “Convenzione universale sul diritto d'autore³” prima e l'“Accordo sugli aspetti commerciali dei diritti di proprietà intellettuale⁴” poi) alcuni dei suoi aspetti sono stati standardizzati e resi validi internazionalmente.

1.4.4 Copyleft



Figura 1.7: Simbolo Copyleft

Con Copyleft si intende una forma di concessione di licenza con cui il detentore dei diritti su un'opera utilizza le leggi sul copyright per concedere agli usufruttori i diritti di modificare e ridistribuire copie dell'opera, con l'obbligo a mantenere questi diritti anche sulle versioni modificate, in modo che l'opera rimanga sempre disponibile al pubblico.

Il concetto di copyleft è stato introdotto da Richard Stallman intorno alla metà degli anni '80, per contrastare il concetto di software proprietario. Pur utilizzando quelle stesse leggi sul copyright che permettevano l'esistenza del software proprietario, Stallman creò una licenza basata sul concetto di copy-

²1886

³1952

⁴1994

left⁵, che usò poi successivamente come base per la licenza GPL. Nell'ambito del software, cioè il contesto in cui il concetto è nato, le licenze copyleft impongono che il codice sorgente venga reso disponibile ai destinatari del programma, così da permettere la trasmissione dei diritti di modifica, e che nel caso questo codice venga modificato e rilasciato, allora ciò deve avvenire sotto la licenza originaria.

Copyleft forte

Una licenza che impone che tutti i software derivati da quello originario (anche nel caso ne venga usata solo una parte) siano rilasciati sotto la sua stessa licenza, così da creare un effetto “virale” che si trasmette a tutte le opere derivate. Questa “viralità” ovviamente non è solitamente gradita ai produttori di software proprietario, in quanto questa rende ovviamente impossibile l'utilizzo del software rilasciato sotto licenza con copyleft forte all'interno di programmi, anche solo in parte closed source, dato che la sua conseguenza è quella di imporre l'applicazione della licenza (e quindi delle sue clausole) all'intero software. Un esempio di licenza con copyleft forte è GPLv3, in quanto contiene la condizione di propagazione della licenza su tutti i programmi che hanno almeno una loro parte originariamente coperta da essa.

⁵Emacs General Public License http://www.free-soft.org/gpl_history/emacs_gpl.html

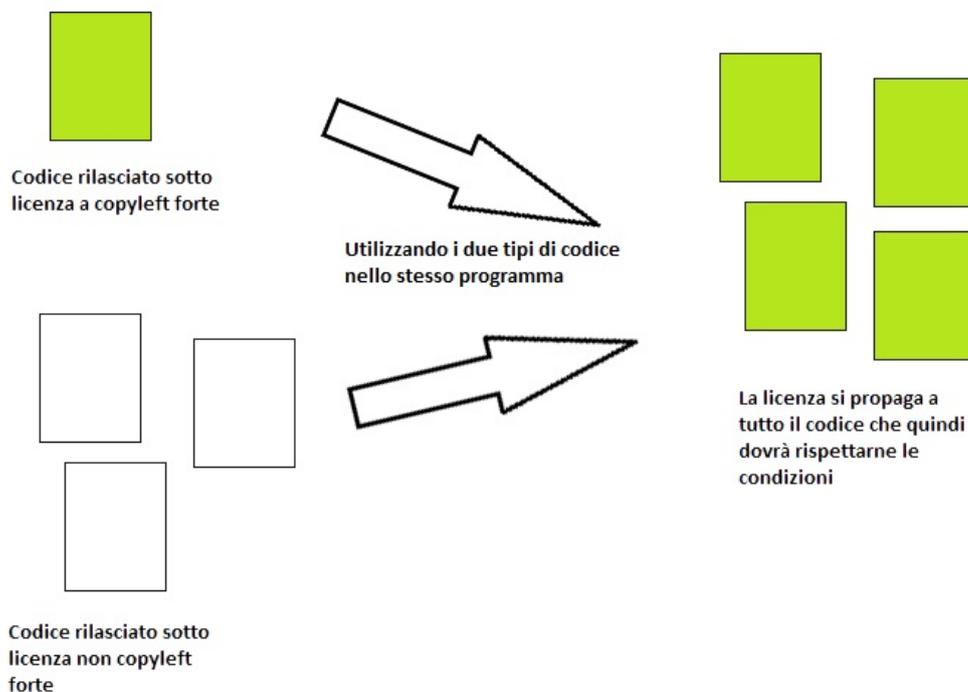


Figura 1.8: “Viralità” della licenza

Copyleft debole

Si definisce così una licenza che impone il suo utilizzo solo su certi tipi di opere derivate, di solito imponendo l’applicazione della stessa licenza sulle versioni modificate del software, ma non su programmi che li utilizzano tramite il linking. Questo tipo di licenze di solito sono utilizzate per le librerie, che quindi possono essere utilizzate da un numero maggiore di programmi e non solo su quelli coperti dalla stessa licenza.

Un esempio di licenza con copyleft debole è LGPLv3, che se applicata a un software, permette il suo utilizzo anche all’interno di software proprietario, in quanto essa impone la pubblicazione dei sorgenti della sola parte coperta dalla licenza.

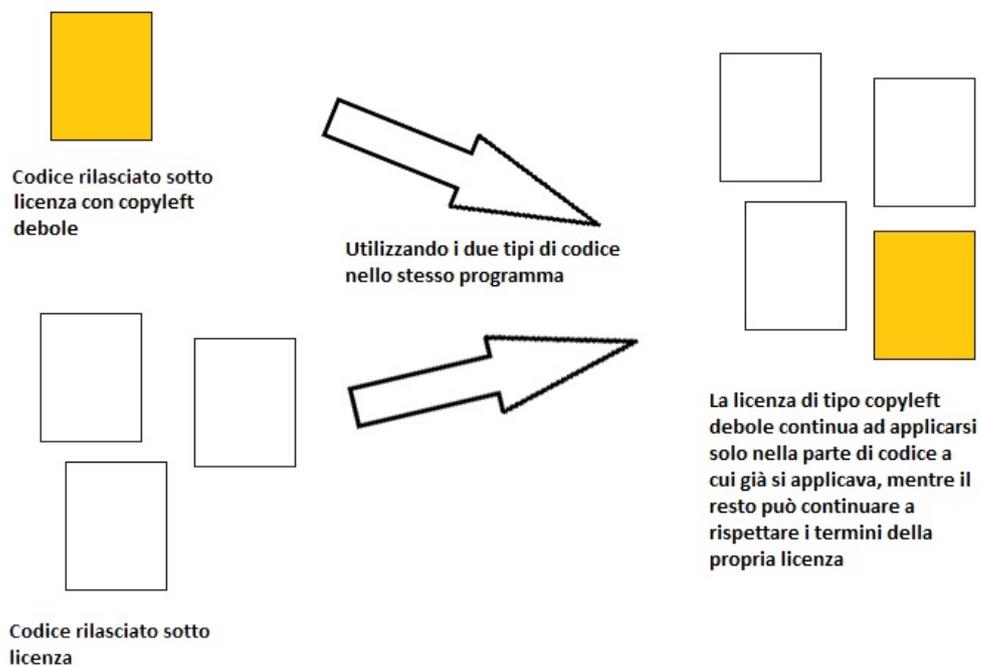


Figura 1.9: “Non viralità” della licenza

Capitolo 2

Scelta della licenza

2.1 GPLv3 riassunta punto per punto

Questa sezione non intende sostituirsi alla lettura approfondita della licenza originale, ma solo aiutare a comprenderne i punti principali elencati nelle sue varie sezioni.



Figura 2.1: Logo della licenza GPL versione 3

La parte iniziale della licenza contiene la nota di copyright che afferma che essa è di proprietà della Free Software Foundation, e la nota che afferma che chiunque volesse utilizzarla per i propri scopi può farlo, ma solo a condizione che essa sia riprodotta senza apportarle alcun cambiamento.

Dopodiché si trova il preambolo in cui viene spiegata la filosofia alla base della licenza e vengono riassunte le sue caratteristiche principali, in esso viene affermato che questa è una licenza libera e copyleft, che può essere applicata a software e altri tipi di lavori e che, al contrario di molte altre licenze, essa

è intesa per garantire la libertà di modifica e di condivisione del programma a cui è applicata.

Viene poi specificato che con il termine “software libero” si intende sia la libertà di distribuire copie di un software libero, volendo anche a pagamento, sia la possibilità di ricevere ed utilizzare il codice sorgente per modificare il programma, o per usarne parti all’interno del proprio programma libero.

In seguito viene spiegato che, quando si distribuiscono copie di un programma bisogna fare in modo di accordare ai destinatari gli stessi diritti ricevuti, il che comprende anche l’accesso al codice sorgente e la conoscenza dei termini della licenza, in modo che coloro che ricevono il programma possano sapere quali sono i loro diritti.

La licenza afferma poi che per proteggere gli autori del programma non viene fornita nessuna garanzia sul software e che eventuali versioni modificate devono essere contrassegnate come tali, in modo che eventuali problemi insorti a causa delle modifiche apportate non vengano attribuiti alla versione originale del programma.

Infine viene affermato che la licenza proibisce la pratica della Tivoizzazione e che le patenti sul software non possono essere utilizzate per rendere il programma non libero.

Termini e condizioni

1. Codice sorgente

Dopo un breve paragrafo dedicato al glossario dei termini utilizzati e la loro interpretazione specifica in questa licenza, viene precisato che il codice sorgente che deve essere reso disponibile deve comprendere tutto ciò che è necessario per generare, installare, eseguire ed eventualmente modificare il programma, compresi eventuali script utilizzati, file di definizione delle interfacce e il codice sorgente per librerie condivise e sottoprogrammi linkati dinamicamente specificamente richiesti. Invece non c’è bisogno di includere librerie di sistema, strumenti general purpose o programmi liberi, che sono liberamente accessibili e che sono

utilizzati senza modifiche nello svolgimento di quelle attività, ma che non fanno parte di quel lavoro e non è necessario includere neppure qualsiasi elemento un utente possa rigenerare da altre parti del codice sorgente.

2. Autorizzazioni di base

In questo paragrafo si afferma che non è possibile revocare nessuno dei diritti concessi dalla licenza finché è rispettata, e che essa non copre l'output prodotto durante l'esecuzione, a meno che esso non contenga parte del codice sorgente.

3. Tutelare i diritti legali degli utenti dalle leggi anti-aggiramento

Con la distribuzione del lavoro a cui è applicata questa licenza si rinuncia a qualsiasi potere legale di vietare l'aggiramento di misure tecnologiche (DRM), se tale aggiramento è effettuato nell'esercizio dei diritti garantiti dalla licenza sul programma, e si rinuncia all'intenzione di limitare l'operatività o la possibilità di modificare il lavoro per far valere i propri diritti legali o quelli di terzi di vietare l'aggiramento di quelle misure.

4. Distribuire copie esatte

È permessa la redistribuzione del codice sorgente in qualsiasi formato, a condizione che in ogni copia sia presente la notifica di copyright appropriata, che sia presente una copia della licenza e che vengano lasciate intatte le notifiche di assenza di garanzia sul software, e quella che attesta l'applicazione della licenza ed eventuali condizioni aggiuntive non permissive (i cui termini verranno specificati nel paragrafo 7 della licenza) al codice.

5. Distribuire versioni modificate

È permessa la distribuzione di versioni modificate del programma, a condizione che sia reso chiaro che si tratta di una versione modificata,

che tutte le note di copyright vengano aggiornate con i dati necessari (quindi nel caso il programma originario avesse una interfaccia utente interattiva che le mostrava, andranno aggiornate anche quelle) e che tutte le sue parti siano licenziate sotto questa licenza. Viene chiarito poi che nel caso in cui un programma faccia parte di un aggregato di software, inteso come un insieme di programmi indipendenti tra loro che si trovano nello stesso supporto (ad esempio un hard disk o un cd con una raccolta di programmi), la licenza di quel software rimarrà confinata a quello specifico software e non contaminerà il resto dell'aggregato.

6. Distribuire in formato non sorgente

Qui viene spiegato in che modo può essere distribuito il programma ed il relativo codice sorgente. In pratica se il software è distribuito tramite un formato fisico, i sorgenti devono essere distribuiti sempre in forma fisica o, nel caso non sia possibile allegare i sorgenti, bisogna che vi sia presente un'offerta scritta di fornire i sorgenti relativi gratuitamente se scaricabili da un server online, o se forniti su un supporto fisico (CD-ROM, DVD, ecc...) di non richiedere più delle spese sostenute necessarie a fornire il servizio. Questa offerta deve essere valida per almeno 3 anni e durare finché quella versione viene supportata.

Se invece il software non è distribuito su un supporto fisico, allora i sorgenti devono essere forniti senza ulteriore ricarico sul prezzo del programma e nello stesso posto. Nel caso essi si trovino su due Server differenti bisogna che siano fornite indicazioni chiare su dove trovare i sorgenti corrispondenti. In questo caso è permesso distribuire il software con l'offerta scritta di fornire il codice sorgente solo occasionalmente e non commercialmente, e solo se il software è stato ricevuto già con quell'offerta inclusa. Volendo si può distribuire anche tramite i peer-to-peer, ma solo come forma supplementare a quella precedentemente indicata e a condizione che siano forniti a tutti i peers infor-

mazioni su dove il software e i sorgenti sono disponibili per il resto del pubblico. Il prerequisito di fornire le informazioni di installazione non implica l'obbligo di dover fornire anche servizi di supporto, garanzie o aggiornamenti sul software. Il codice sorgente e le informazioni di installazione fornite devono essere in un formato documentato pubblicamente e si deve fare in modo che per scompattarli, leggerli e copiarli non sia necessario l'uso di password o chiavi di decrittazione.

7. Termini addizionali

Lo scopo principale di questa sezione è quello di aumentare la compatibilità tra questa e altre licenze. L'inserimento di termini addizionali può essere applicato aggiungendoli direttamente a quelli già presenti all'inizio dei relativi file sorgente a cui vanno applicati, oppure aggiungendovi una notifica su dove trovarli. È permesso l'inserimento di autorizzazioni addizionali sotto forma di eccezioni a una o più condizioni dei termini della licenza se esse sono valide sotto le normative vigenti. Nel caso queste eccezioni si applichino solo a parte del programma, allora quella parte può essere utilizzata secondo le loro condizioni, ma l'intero programma rimane sottoposto a questa licenza senza considerare ulteriori permessi. Questi permessi possono essere comunque rimossi dalle copie che vengono distribuite.

Nel caso si aggiunga materiale al software distribuito si possono integrare i termini alla licenza su quel materiale (se si ha il permesso del/si è il detentore del Copyright di quel materiale) con termini che:

- Declinano la garanzia o limitano la responsabilità in maniera differente da questa licenza
- Richiedano la preservazione di specifiche note legali o l'attribuzione degli autori, nel materiale o nelle note legali, di lavori che lo contengano.

- Proibiscano il travisamento dell'origine del materiale, e richiedano che versioni modificate di esso vengano indicate come differenti dall'originale.
- Limitino l'utilizzo del nome dei detentori della licenza o degli autori per scopi pubblicitari.
- Rifiutino di concedere i diritti per alcuni nomi commerciali, marchi di fabbrica o marchi di servizio
- Richiedano l'indennizzo dei detentori del copyright o degli autori, nel caso il materiale venga distribuito con impegni contrattuali che impongano responsabilità su di essi, questo indennizzo pesa su colui che ha distribuito il materiale (o una sua versione modificata) con responsabilità nei confronti del destinatari.

Altri tipi di termini restrittivi vengono invece considerati “ulteriori restrizioni” e quindi, secondo la sezione 10 di questa licenza, non possono essere imposti e il ricevente può rimuoverli.

8. Terminazione

Ogni tentativo di propagare o modificare un lavoro coperto da questa licenza senza rispettarne i termini viene considerato nullo e causa la terminazione automatica di tutti i diritti concessi dalla licenza, ma coloro a cui sono state distribuite copie coperte dalla licenza non ne vengono colpiti.

Una volta ricevuta la notifica di violazione da parte del detentore del copyright, se questa è la prima infrazione alla licenza di un'opera di quel proprietario e se si risolve il problema entro 30 giorni dalla ricezione della notifica, allora la licenza viene ripristinata. In generale se ci si accorge di stare infrangendo la licenza e si risolvono le violazioni, a meno che il detentore del Copyright non la termini permanentemente, la licenza viene ripristinata automaticamente in maniera temporanea, se poi sono passati più di 60 giorni dalla risoluzione e non si sono

ricevute notifiche dal detentore del Copyright, allora questo ripristino diventa permanente.

9. L'accettazione della licenza non è richiesta per l'ottenimento di copie

Dato che questa licenza copre i diritti di modifica e distribuzione, se si usa software coperto da questa licenza senza compiere una di queste due azioni non c'è l'obbligo di accettarne i termini. Se però si modifica o distribuisce il software, significa che si sono accettati tutti i termini della licenza.

10. Concessione automatica della licenza ai destinatari successivi

Ogni volta che un programma coperto dalla licenza viene distribuito a qualcuno, questi riceve la licenza direttamente dal licenziante originario. Non si possono imporre ulteriori restrizioni, né richiedere ulteriori pagamenti per poter esercitare i propri diritti concessi dalla licenza, né intentare cause per violazione di brevetto a coloro che utilizzano il programma.

11. Brevetti

Ognuno dei detentori del copyright concede una licenza sui suoi brevetti non esclusiva, mondiale e libera da royalty, che permette di usare, modificare e distribuire quella parte del programma. Nel caso si voglia usare materiale coperto da brevetto all'interno di un proprio lavoro, lo si potrà fare solo se anche quella parte può essere distribuita seguendo tutti i requisiti della licenza.

12. Nessuna rinuncia delle libertà altrui

Nel caso non sia possibile distribuire il software rispettando la licenza e contemporaneamente eventuali altri obblighi (come per esempio accordi o ordinanze del tribunale), allora non si può distribuire il proprio lavoro sotto questa licenza.

13. **Utilizzo insieme alla GNU Affero General Public License**

Viene concesso il permesso di combinare ogni lavoro coperto da questa licenza con uno coperto dalla AGPL versione 3, nonostante quanto affermato da qualsiasi altra clausola di questa licenza. In tal caso questa licenza continuerà a coprire le parti originarie, mentre la GNU Affero General Public License coprirà la loro combinazione.

14. **Versioni aggiornate di questa licenza**

Nel caso nel programma venga specificato che a esso viene applicata una certa versione della licenza GNU General Public License, “o qualsiasi versione successiva”, allora si può scegliere di seguire le condizioni imposte da quella versione o qualsiasi versione successiva. Questo comporta il fatto che se una versione successiva dovesse imporre ulteriori vincoli, quella versione del software non è obbligata a rispettarli.

15. **Rinuncia alla garanzia, 16.Esonero dalle responsabilità e 17.Interpretazione delle sezioni 15 e 16**

Il programma non è provvisto di garanzia e qualora il suo utilizzo dovesse provocare qualsiasi tipo di danno, i detentori del copyright e i distributori di quel programma non ne potranno essere ritenuti responsabili. Qualora le leggi locali impedissero l'applicazione della rinuncia di responsabilità, allora bisognerà applicare la legge che più si avvicina alla rinuncia alle responsabilità civili connesse al programma.

Nella parte finale è presente un breve tutorial su come applicare la licenza al proprio programma, che qui però non verrà riassunto, in quanto nel prossimo capitolo di questo elaborato verrà inserito un tutorial comprendente anche questo argomento.



Figura 2.2: Logo della licenza AGPL versione 3

2.2 Breve descrizione AGPLv3 (GNU Affero General Public License Versione 3)

La licenza Affero versione 3 è una licenza a copyleft forte che è stata pubblicata nel 2007 dalla Free Software Foundation ed è stata riconosciuta come licenza open source nel 2008¹ dalla Open Source Initiative. Questa è una versione più restrittiva della GPLv3, in quanto invece di imporre la diffusione del codice sorgente solo quando il software viene distribuito, essa impone di renderlo disponibile anche nel caso il software sia solo eseguibile tramite una rete.

Essenzialmente la licenza GPLv3 e la licenza AGPLv3 sono identiche, tranne che per la sezione 13 che nella licenza GPLv3 afferma che è possibile combinare un lavoro coperto da essa con uno coperto dalla AGPLv3, e pur essendo essa applicata a quel lavoro, permette di applicare anche la restrizione aggiuntiva imposta dalla AGPLv3, ovvero di fornire la possibilità di ottenere il codice sorgente del programma anche a tutti coloro che lo utilizzano remotamente tramite una rete di computer, rendendo di fatto le due licenze compatibili. Questa licenza risolve quello che da alcuni era considerata una mancanza della GPLv3, che da sola permette a certe aziende di utilizzare e modificare software open source per la creazione di applicazioni web, che vengono utilizzate dall'utente solo remotamente, senza che queste aziende siano obbligate a fornire i sorgenti, sfruttando quello che viene chiamato "ASP loophole" e andando così contro al concetto di software libero.

¹http://www.funambol.com/news/pressrelease_2008.3.13.php

2.3 Breve descrizione LGPLv3 (GNU Lesser General Public License versione 3)



Figura 2.3: Logo della licenza LGPL versione 3

La licenza LGPLv3 è una versione basata sulla licenza GPLv3, in quanto essa applica le stesse condizioni della licenza madre, tranne nei casi compresi dalle eccezioni che la compongono, essa è quindi considerata a copyleft debole.

La licenza è stata pubblicata nel 2007 dalla Free Software Foundation e successivamente nello stesso anno è stata approvata come licenza open source² dalla Open Source Initiative. In pratica è simile alla licenza GPLv3 ma con una serie di permessi aggiuntivi che la rendono meno restrittiva, infatti al contrario di questa, non impone che essa venga trasferita automaticamente al resto del programma che utilizza la libreria o il software a cui è stata applicata. Questa licenza è stata creata principalmente per permettere agli sviluppatori di software di utilizzare librerie libere all'interno dei loro programmi, sia che siano considerati software libero che no, ma in ogni caso il codice sorgente della parte del software coperta dalla licenza LGPLv3 deve essere condiviso.

2.4 Principali differenze tra GPLv2 e GPLv3

Al momento del rilascio della versione attuale (2007) la licenza open source più utilizzata era la GPL versione 2, che fu rilasciata nel lontano 1991. Alcuni cambiamenti sono stati apportati alla versione 3, i principali dei quali sono spiegati in seguito. Alcuni di questi cambiamenti hanno contribuito ad

²<http://opensource.org/node/193>

alimentare la controversia sorta tra i sostenitori della seconda versione, che vedono quella nuova come inutilmente più restrittiva, e quelli della terza che invece ritengono i cambiamenti necessari per adattare la licenza alle nuove necessità sorte nel frattempo, e contemporaneamente per proteggere quelli che sono percepiti come i principi fondamentali del software libero. Qualunque sia la propria visione sull'argomento, le due versioni sono entrambe ancora valide ed utilizzabili per il proprio software, quindi l'utilizzo di una o l'altra è più una questione di preferenze (a meno che non si utilizzi anche software già coperto da una specifica licenza, in tal caso bisogna anche considerare la compatibilità tra di esse).

Questa nuova versione sta man mano conquistando una fetta sempre più ampia del software libero e open source, ma anche a causa del fatto che essa è stata rilasciata molto più recentemente e per la difficoltà che presenta il cambio di licenza dalla versione precedente a quella attuale per i progetti con svariati collaboratori, essa è ancora meno utilizzata della versione 2.

Per cominciare, nel passaggio dalla versione 2 alla 3 il linguaggio del testo è stato fortemente modificato in modo da renderlo più inclusivo, ordinato e internazionale e per adattarlo ai cambiamenti legislativi che sono sorti negli anni trascorsi tra il rilascio delle due licenze, facendo ciò si è però reso il testo della licenza molto più lungo e difficile da interpretare per chi non è un esperto nell'argomento.

La nuova versione della licenza è stata resa più compatibile con altre licenze libere (tra cui Apache v2.0 e la Affero General Public License versione 3) ed è stato chiarificato il caso specifico in cui il licenziatario concede l'accesso al lavoro solo sotto forma di servizio ospitato su un host (quindi senza trasferimento di copie del lavoro), che non viene considerato trasmissione, e quindi non viene posto l'obbligo di fornire il codice sorgente (se si applica la AGPL invece bisogna fornirlo). Sono state aggiunte le cosiddette misure contro la "tivoizzazione", che viene tollerata solo per programmi utilizzati da aziende o organizzazioni, inoltre è stato chiarificato il rapporto con i brevetti, affermando che se il possessore del brevetto sul software rilascia il software

sotto licenza GPLv3, esso rinuncia alla rivendicazione dei diritti sui propri brevetti nei confronti dei licenziatari del programma (ad esempio non può imporre royalty o intentare causa agli utilizzatori o distributori del programma per violazione di brevetto), altrimenti gli verrà revocata la licenza GPLv3 e tutte le licenze sui brevetti relativi. È stato anche specificato che è permesso usare la licenza su codice che implementa DRM³, ma se qualcuno riesce a rimuoverli, può redistribuirlo senza di essi e senza che il proprietario possa invocare eventuali leggi statali per impedirlo.

Un grosso cambiamento è avvenuto anche nella gestione delle violazioni della licenza, in quanto nella versione 2 della licenza coloro che ne infrangono i termini perdono automaticamente il loro diritto di modifica e distribuzione, per cui anche se a un certo punto risolvono la violazione senza che il proprietario del copyright abbia notificato il problema, essi devono comunque contattarlo per richiedere il ripristino. Questo ovviamente implica seri problemi per coloro che infrangono la licenza di un programma in maniera involontaria, in quanto devono sperare che il proprietario sia rintracciabile e disposto a ripristinarla. Nella versione 3 si è deciso invece che se i trasgressori risolvono la violazione e non vengono contattati dal proprietario del copyright entro 60 giorni dal momento della correzione, o se essi risolvono il problema entro 30 giorni dalla ricezione della prima notifica di infrazione del copyright, il ripristino permanente dei diritti è automatico, permettendo quindi al contravventore volenteroso di poter tornare ad utilizzare il software anche nel caso il contatto con il proprietario fosse impossibile. Un altro cambiamento importante apportato alla licenza è stato l'aggiunta di una sezione in cui viene indicato il suo rapporto con eventuali permessi aggiuntivi, quali tipi di termini addizionali è permesso aggiungere al proprio lavoro, e come devono comportarsi nei loro confronti gli eventuali distributori.

³nella licenza tradotto in Digital Restrictions Management

2.4.1 Tivoizzazione

Il termine deriva da “TiVo”, che è il nome dato a una serie di videoregistratori digitali diffusi principalmente negli Stati Uniti d’America, il cui software che ne permetteva il funzionamento era basato su Linux, un sistema operativo licenziato sotto GPL, e quindi anch’esso era obbligato a sottostare alla stessa licenza. A partire dal rilascio della seconda versione (serie 2) in esso vennero incluse delle protezioni hardware che ne impedissero l’utilizzo con software modificato (in conformità alle leggi federali U.S.A. sul supporto alla TV via cavo), e quindi pur non infrangendo direttamente le regole della GPLv2 (dato che comunque la ditta produttrice metteva a disposizione del pubblico tutti i sorgenti, ma questi sorgenti non potevano essere usati per creare versioni alternative del software che potessero essere utilizzate con quell’hardware specifico), questa limitazione venne vista come fortemente contraria allo spirito del software libero. Per contrastare ciò, nella versione 3 della licenza GPL, venne introdotta la seguente nota nel paragrafo 6, per impedire che in futuro altro software libero coperto dalla licenza venisse “sfruttato” in questo modo:

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the abi-

lity to install modified object code on the User Product (for example, the work has been installed in ROM).

Capitolo 3

Applicazione della Licenza e distribuzione del programma

In questo capitolo esamineremo il programma XCMoel nella sua versione più recente, per poi provvedere a mostrare quali sarebbero i passi necessari per prepararlo per il rilascio nel mondo dei FOSS¹, infine verranno forniti alcuni consigli su certi argomenti che si ritiene utile conoscere.

3.1 XCMoel un sistema per modellare curve e superfici NURBS

XCMoel è un sistema grafico interattivo di tipo CAD realizzato in ambito accademico, per creare o modificare curve e superfici, in modo da realizzare oggetti solidi e fare il rendering delle scene che li contengano.

Esso è il risultato (in continua evoluzione) di svariati anni di ricerca nel campo della computer grafica, e in cui vengono prima sperimentate e poi eventualmente implementate svariate tecniche e algoritmi per la modellazione e il rendering di vari oggetti sia 2D che 3D.

¹Free and Open Source Software



Figura 3.1: Schermata XCMoDel

Questo significa che esso non è solo utilizzabile come mezzo didattico, ma può essere impiegato anche per scopi professionali, grazie anche ad alcune caratteristiche che sono allo stesso livello di software di tipo commerciale simili. Per realizzare ciò esso integra vari componenti più o meno specializzati in uno specifico campo e che vengono utilizzati per i seguenti scopi:

- **xccurv**: utilizzato per la modellazione di curve 2D di tipo NURBS.

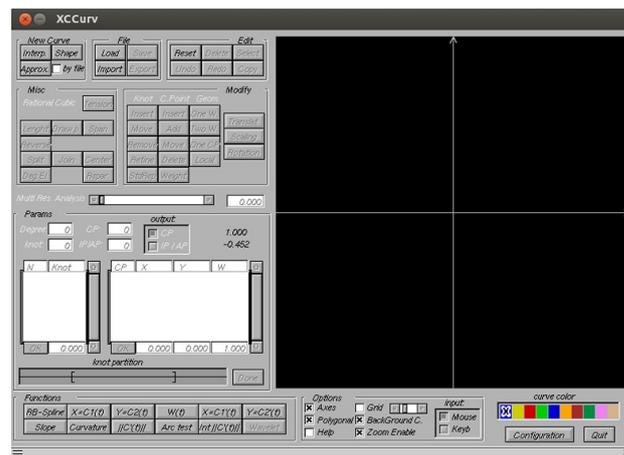


Figura 3.2: Schermata XCCurv

Queste curve possono essere create sia per l'inserimento dei punti di controllo, che per interpolazione, o tramite approssimazione, e posso-

3.1 XCMoel un sistema per modellare curve e superfici NURBS 31

no in seguito anche essere modificate, sia applicando le tradizionali trasformazioni di traslazione, scala e rotazione, sia agendo sui loro vari parametri (ad esempio aggiungendo, spostando o eliminando i nodi e/o i punti di controllo). Alcune funzionalità interessanti che questo pacchetto implementa al momento sono l'utilizzo di classi di interpolazione locale razionale per curve, un prototipo di MRA² basato su funzioni wavelet non uniformi e la possibilità di importare o esportare curve in formato .igs (IGES³).

- **xcsurf**: utilizzato per la modellazione di curve e superfici 3D di tipo NURBS.

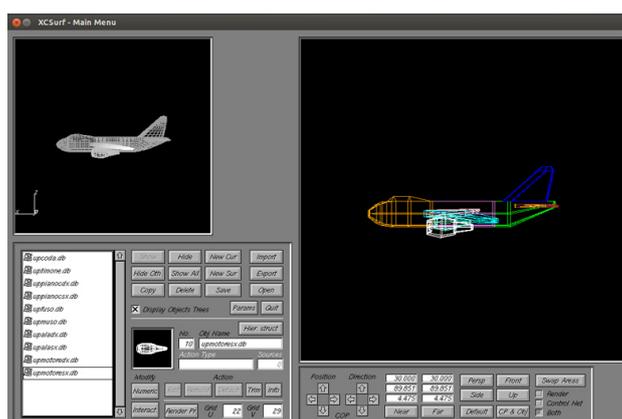


Figura 3.3: Schermata XCSurf

A queste possono essere applicate le trasformazioni tradizionali, sia interattivamente tramite il mouse, che numericamente tramite la tastiera; e in più è possibile creare superfici applicando le tecniche di estrusione, skinning, rivoluzione e swinging a varie curve opportunamente posizionate. Questo pacchetto è stato realizzato con l'intenzione di renderlo sufficientemente performante pur compiendo tutte queste operazioni via software, in modo che esso possa essere utilizzato agevolmente anche

²Multiresolution analysis

³Initial Graphics Exchange Specification: è utilizzato come formato di scambio dati fra sistemi di tipo CAD

su macchine sprovviste di specifiche risorse hardware. Esso implementa svariate funzionalità molto utili per l'utente che ne fa uso. Alcuni esempi sono l'utilizzo dell' "albero di costruzione" per mantenere la storia delle operazioni effettuate, il "ricalcolo automatico" per permettere una progettazione dinamica, i plug-in per la "Free Form Deformation" e quello per la modellazione gerarchica, e molteplici altri.

- **xcboul:** utilizzato per l'esecuzione di operazioni booleane tra solidi (intersezione, unione, ecc...).

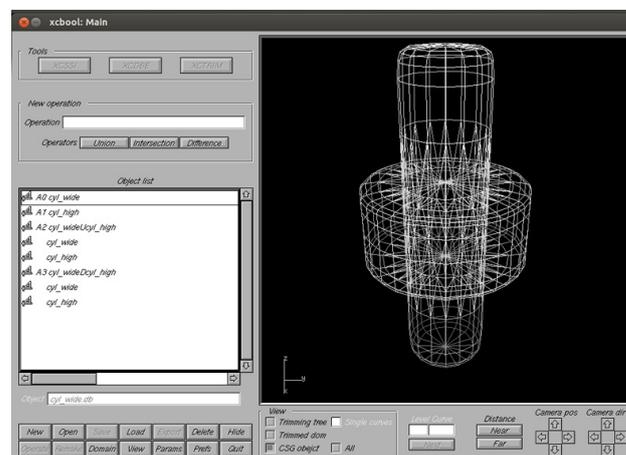


Figura 3.4: Schermata XCBoul

È stato uno dei primi pacchetti implementati, e pur non avendo subito modifiche, utilizza un algoritmo estremamente complesso di composizione booleana tra superfici nurbs.

- **xcrayt:** utilizzato per la rappresentazione di una scena (o di un'animazione composta da un insieme di scene) e il suo rendering fotorealistico tramite un algoritmo di ray tracing.

3.1 XModel un sistema per modellare curve e superfici NURBS 33

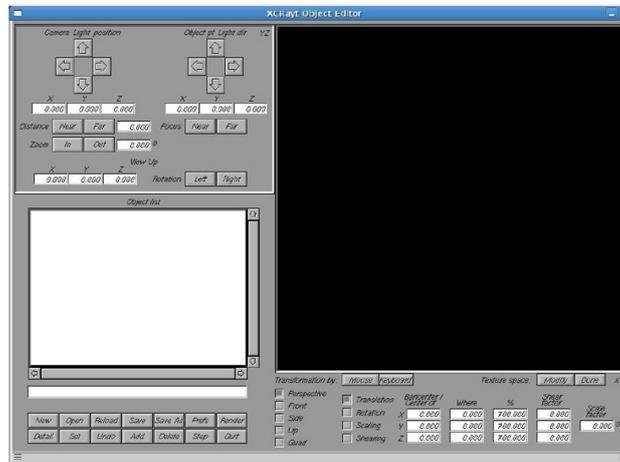


Figura 3.5: Schermata XCRayt

Queste possono essere create sia in modo interattivo, direttamente all'interno dell'ambiente di XCRayt, sia tramite l'utilizzo di script in linguaggio C. Per quanto riguarda il rendering, esso utilizza il pacchetto hrayt che implementa diversi algoritmi di ray tracing puro, che utilizzano l'intersezione raggio-superficie, contrariamente ai maggiori motori di ray tracing che invece approssimano le superfici con delle mesh e poi procedono all'intersezione raggio-piano.

- **xmovie (xcview):** utilizzato per visualizzare la versione renderizzata della scena (o dell'animazione), per esportarla in formato .avi o .gif, o per convertire le rese in un formato diverso da quello prodotto da XCRayt.

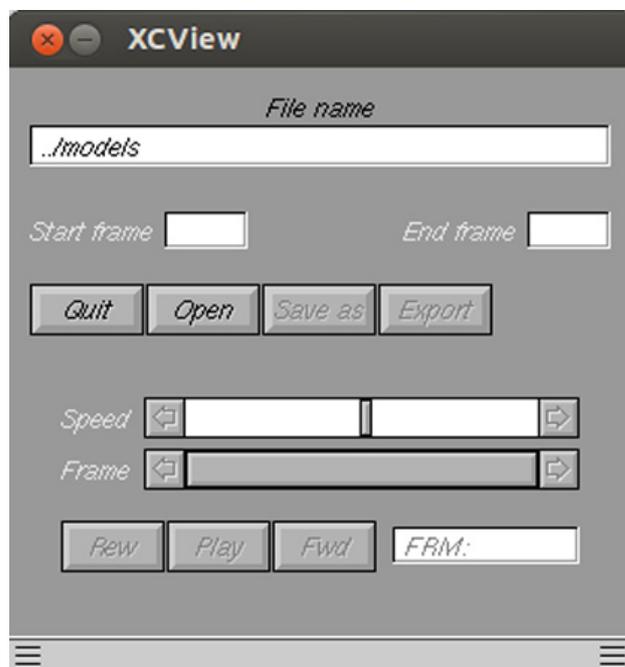


Figura 3.6: Schermata XCVIEW

- **xctrim:** pacchetto ancora in via di sviluppo che una volta completato sarà un potente strumento per la gestione delle superfici trimmate.
- **XTools:** è una libreria che serve a fornire i mezzi necessari per creare facilmente interfacce grafiche in un ambiente Xwindow, invece di utilizzare direttamente gli strumenti messi a disposizione da esso, in quanto questi sarebbero più complicati del necessario da utilizzare per gli scopi del progetto. Nonostante tutto, questa libreria è molto efficiente e veloce, e in più non richiede l'utilizzo di molta memoria.
- **Trim:** questa libreria grafica permette il rendering in tempo reale via software di superfici NURBS sia trimmate che non, in modalità differenti (wireframe, hidden line, depth cueing, o con shading a scelta tra Gouraud e Phong). La visualizzazione in tempo reale è permessa dalla tassellatura (tessellation⁴) della superficie che la approssima in

⁴generazione automatica scalare di poligoni su superfici piane e non

3.1 XCMoDel un sistema per modellare curve e superfici NURBS 35

una mesh 3D, e ciò permette di guadagnare velocità a discapito della precisione della rappresentazione.

- **Descriptor:** lo scopo di questa libreria è quello di descrivere una scena 3D tramite un programma scritto in C, permettendo ad esempio di assegnare determinate caratteristiche (come il colore, la posizione, le dimensioni, ecc...) ai vari oggetti che la compongono, e nelle sue ultime versioni permette anche la descrizione di piccole animazioni. Una volta definita la scena sarà poi possibile visualizzarla, ed eventualmente farne il rendering in qualsiasi momento, tramite il pacchetto XCRayt.
- **Matrix:** è una libreria di funzioni per facilitare la gestione delle operazioni su vettori e matrici, ed è utilizzata principalmente quando i vari pacchetti devono attuare trasformazioni o proiezioni geometriche.

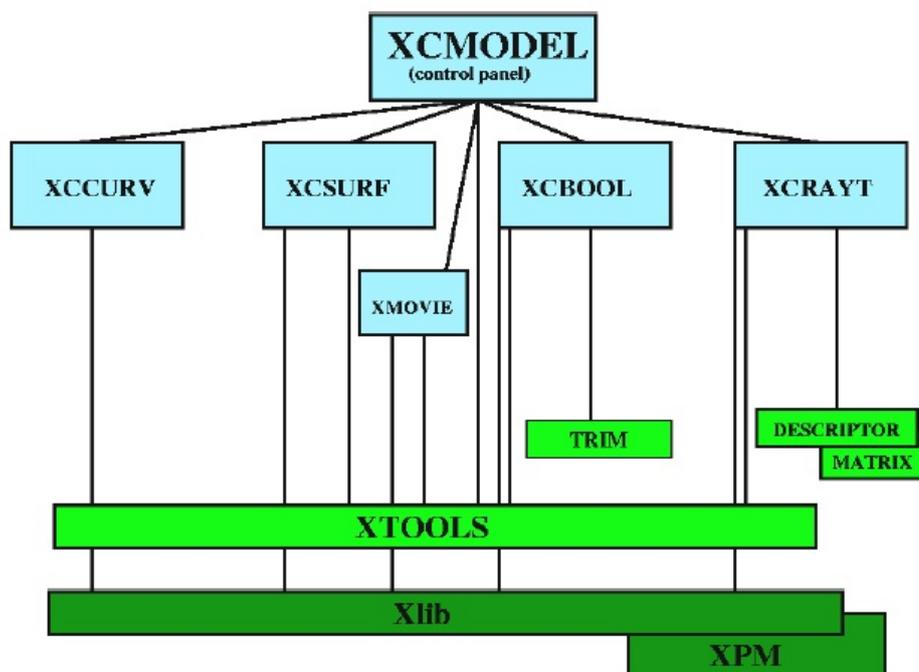


Figura 3.7: Struttura XCMoDel

Per convertire un'animazione dal formato .hra (che è un file composto da un elenco di file .hr ognuno contenente un fotogramma) a .avi o .gif, il pacchetto xview attualmente utilizza alcuni programmi esterni. Per la conversione in .gif animata, prima di tutto provvede a convertire tutti i fotogrammi dal formato .hr al formato .ppm, dopodiché genera uno script che, quando eseguito, converte i file dal formato .ppm al formato .gif e che poi avvia “gifmerge” che provvederà ad unire questi ultimi in un unico file .gif animato, secondo i parametri che gli verranno passati. Per convertire l'animazione nel formato .avi, dopo aver convertito i vari frame nel formato .ppm come nel caso precedente, genera un altro script che converte da questo formato al formato .jpg e che poi avvia “jpegtoavi”, che provvederà a trasformare i singoli frame contenuti nei file .jpg in un unico file .avi contenente l'animazione (anche in questo caso secondo i parametri scelti dall'utente). Perché il tutto funzioni correttamente bisogna però che i seguenti pacchetti siano stati installati:

- **Netpbm:** è un insieme di applicazioni e librerie per la conversione tra formati file di immagini, che permette anche di eseguire su di esse operazioni come il ridimensionamento, il ritaglio delle immagini, ecc... Xcview lo utilizza per convertire le immagini nei formati richiesti da “jpegtoavi” e “gifmerge” e cioè dal formato .ppm a .jpg o .gif.
- **FFmpeg:** è un insieme di applicazioni e librerie per la codifica e la decodifica di file in svariati formati video, che viene utilizzato tramite riga di comando. In questo caso specifico è necessario averle installate se si vuole che il filmato sia esportato in un .avi con codec Mpeg4.

XCModel è stato progettato per funzionare su un dispositivo con un sistema operativo basato su Linux, ed è stato migliorato nel corso degli anni dal professor Giulio Casciola del dipartimento di Matematica dell'Università di Bologna in collaborazione con altre persone tra le quali anche alcuni suoi studenti. È stato sviluppato in linguaggio C con l'intenzione di renderlo il più

efficiente possibile nelle operazioni di calcolo (che in questo tipo di programmi tendono ad essere molto numerose) e il suo obiettivo dichiarato è quello di fornire un software che non dipenda da librerie commerciali, o comunque non trasportabili su altre piattaforme, che sia indipendente dall'hardware su cui viene utilizzato, e che sia anche facilmente modificabile.

Viene usato abitualmente come mezzo per insegnare praticamente i rudimenti della grafica 3D, durante lo svolgimento delle lezioni di Grafica dei corsi di Informatica e di Matematica dell'Università di Bologna.

Sin dalla sua prima versione risalente al 2000 esso era già distribuito liberamente, ma venivano distribuiti i sorgenti delle sole librerie "xtools", "trim" e "descriptor", mentre i vari pacchetti venivano solitamente distribuiti sotto forma di file binari, e al tutto non era applicata nessuna licenza.

Si è pensato quindi che un eventuale ingresso ufficiale nel mondo dei software liberi o open source del programma, possa portargli considerevoli benefici.

3.1.1 Storia di XCMoel

Il primo pacchetto ad essere stato sviluppato è XCCurv, la cui prima versione risale al 1991 e successivamente, nel 1993 viene creato XCSurf che inizialmente gestisce solo le superfici NURBS, mentre in seguito anche le superfici NURBS trimate.

Nel 1996 viene poi sviluppato un algoritmo efficiente per la gestione delle intersezioni tra superfici NURBS, che successivamente servirà come base per la realizzazione del pacchetto XCBool, e in quello stesso anno avviene la creazione di XCRayt e del suo algoritmo di ray tracing.

Nel 1998 viene deciso di avviare il progetto XCMoel con l'obiettivo di incorporare i vari pacchetti in un unico sistema, e il 20 Gennaio del 2000 ne viene rilasciata la prima versione in occasione della sua presentazione alla conferenza "Analisi Numerica: Metodi e Software Matematico" tenutasi presso l'Università di Ferrara.

La seconda versione di XCMoel esce nel Dicembre dello stesso anno e viene esibita durante la mostra intitolata “Matematica, Arte e Tecnologia: da Escher alla computer graphics” organizzata dal Dipartimento di Matematica dell’Università di Bologna.

Dopo aver corretto numerosi bug, apportato numerose modifiche ai vari pacchetti, e aver aggiunto la possibilità di importare ed esportare curve e superfici nel formato IGES, nel Luglio 2002 viene rilasciato XCMoel 3.0 e presentato all’EGIT’03⁵.

XCMoel 4.0 venne rilasciato pubblicamente nel Luglio 2007 e oltre alla correzione di ulteriori bug trovati nelle versioni precedenti, vengono implementati svariati miglioramenti ai vari pacchetti, tra cui la possibilità di esportare una superficie in formato OBJ Wavefront e l’introduzione della facoltà di creare brevi animazioni in 3D ed eventualmente di convertirle in una GIF animata.

Attualmente è in via di sviluppo la versione 4.3 di XCMoel, che si ripromette di migliorare ulteriormente i vari pacchetti che lo compongono e di eliminare eventuali problematiche insorte nel frattempo.

Il problema maggiore che questa versione ha dovuto risolvere è quello del cambiamento della gestione del backing store da parte di X.Org, che impediva la corretta visualizzazione delle finestre e dei componenti dei vari pacchetti sui computer che avevano le versioni più recenti della loro implementazione del sistema grafico X Window System.

3.1.2 Gestione del progetto e ostacoli

La gestione di questi pacchetti richiede un grande sforzo per mantenerli competitivi nel passare degli anni, a causa dei grandi cambiamenti che continuano ad avvenire nel mondo dell’informatica. Un esempio delle principali modifiche che è stato necessario apportare è avvenuto quando Xorg ha sostituito XFree86 come principale sistema grafico X Window System di tipo open

⁵Eurographics Italian Chapter Conference, Milano

3.1 XCMoDel un sistema per modellare curve e superfici NURBS 39

source per la gestione dell'interfaccia grafica, e quando si è dovuto passare da Imakefile a Cmake per la compilazione e il linking dei pacchetti.

Per favorire la reperibilità dei pacchetti e delle informazioni necessarie per usarli, è stato creato un sito web dedicato esclusivamente al progetto, che solitamente permette di scaricare la versione più aggiornata dei vari pacchetti, e fornisce risorse e tutorial utili per il loro utilizzo. Esso richiede un grande sforzo per essere mantenuto aggiornato ed utile, particolarmente in occasione del rilascio di ogni nuova versione di XCMoDel.

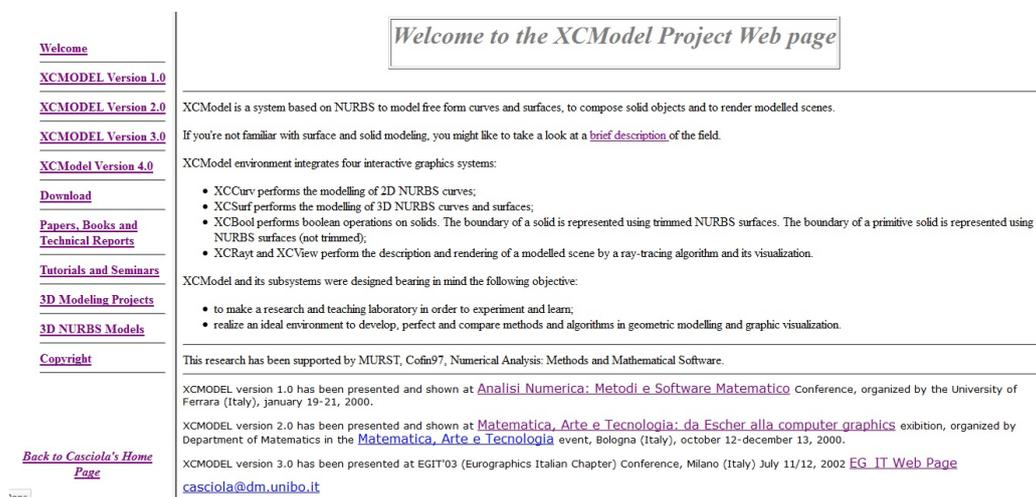


Figura 3.8: Sito ufficiale di XCMoDel

Le sezioni più impegnative da mantenere aggiornate sono le seguenti:

- **Download:** Questa sezione oltre che permettere di scaricare la versione più recente di XCMoDel contiene anche la documentazione dei vari pacchetti.

Questa parte richiede un imponente lavoro per mantenere i manuali per l'utente aggiornati alle ultime versioni dei vari pacchetti, e in più necessiterebbe anche di manuali per lo sviluppatore che sono fondamentali per permettere a chi potrebbe essere interessato a contribuire con lo sviluppo dei vari pacchetti, a farlo in maniera coerente con le scelte fatte fino ad ora.

- **Tutorials and Seminars:** Contiene alcuni tutorials su come utilizzare alcune funzionalità di certi pacchetti. Questa sezione esigerebbe un ampliamento, in quanto il numero di tutorial in essa presenti è molto limitato, e non copre tutte le aree di utilizzo della suite di programmi.
- **3D Modeling Projects:** Qui è possibile trovare alcune delle scene costruite dagli studenti di Grafica in qualità di progetti di fine corso, e le loro relazioni che spiegano come queste scene e gli oggetti che le compongono sono stati costruiti. Queste scene sono suddivise a seconda dell'anno accademico in cui sono state assegnate, e prima di poter essere caricate sul sito richiedono un grande lavoro di selezione e di preparazione. Questa fase richiede il controllo della presenza di tutti i file necessari e l'eliminazione di quelli che non lo sono, la verifica che gli oggetti e le animazioni siano corretti e in più, nel caso durante la fase di controllo dovessero venire alla luce dei problemi, bisogna provare a risolverli, in modo che quando saranno caricati sul sito essi possano essere utilizzati facilmente. L'inclusione delle relazioni degli studenti rende la sezione molto utile, in quanto rimpiazza almeno in parte la necessità di includere dei tutorial, dato che queste permettono di farsi un'idea su come costruire i vari oggetti presenti nelle scene.

The screenshot displays the '3D Modeling Projects' section of a website. On the left is a vertical navigation menu with links for 'Welcome', 'XCMODEL Version 1.0', 'XCMODEL Version 2.0', 'XCMODEL Version 3.0', 'XCMODEL Version 4.0', 'Download', 'Papers, Books and Technical Reports', 'Tutorials and Seminars', '3D Modeling Projects', '3D NURBS Models', and 'Copyright'. The main content area features a title '3D Modeling Projects' in a box. Below it is a 'Select year:' dropdown menu with options from 2000 to 2010 and an 'Other' link. A horizontal line separates this from the 'Year 2010: 4 pic(s)' section. Below this, there is a 'Page 1/1' indicator and a 'HOWTO' link. Three project thumbnails are shown, each with a 'Tutorial' link, a 'Download Project' link, and the file size in Kb: '313' (1399 Kb), 'lamborghini' (2994 Kb), and 'maggiolino' (24575 Kb).

Figura 3.9: Sezione 3D Modeling Projects

3.2 Preparare un pacchetto per la distribuzione con licenza GPLv341

- **3D NURBS Models:** È un archivio di modelli di oggetti suddivisi per categorie e presi prevalentemente dai progetti degli studenti. Come per la sezione precedente, essa richiede che vengano effettuati molti dei controlli necessari per la sezione precedente su ogni oggetto che si voglia rendere disponibile agli utenti, e in più è necessario che vengano effettuate delle operazioni per renderlo utilizzabile anche in altre scene, tra qui per esempio cambiarne le dimensioni in modo che siano proporzionate a quelle degli altri oggetti già presenti nel database.

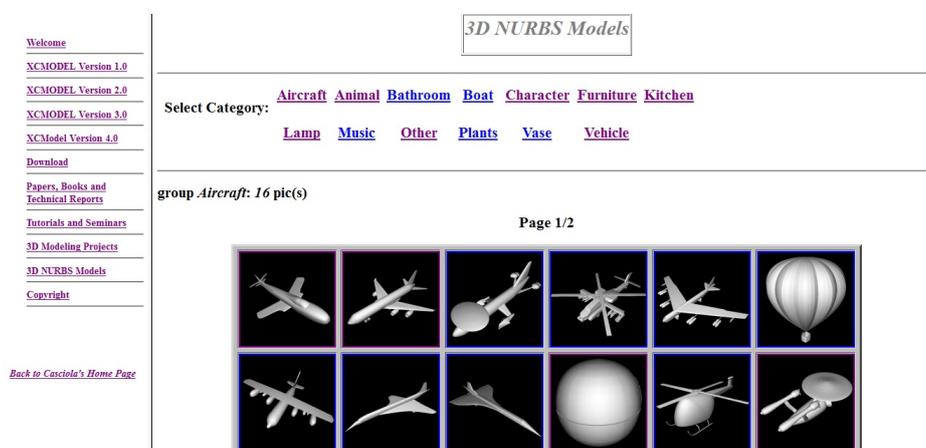


Figura 3.10: Sezione 3D NURBS Models

3.2 Preparare un pacchetto per la distribuzione con licenza GPLv3

Per rendere il proprio programma conforme alle specifiche richieste dalla licenza GPLv3, bisogna prima di tutto verificare se esso rispetti tutti i termini della licenza, una volta confermato questo prerequisito si può procedere con la seguente procedura:

Si inizia includendo una copia della licenza in versione testuale (in lingua originale) all'interno della distribuzione, possibilmente nella cartella superiore del codice sorgente, solitamente in un file di testo. Dopodiché bisogna

aggiungere all'inizio di ogni file sorgente del programma una nota di copyright, che contenga l'anno in cui si è finito di preparare il programma per la distribuzione e il nome degli autori. Nel caso l'anno di preparazione e quello di rilascio non coincidano, allora bisognerà utilizzare quello del primo caso. Nell'eventualità che siano state rilasciate più versioni del programma, allora bisognerà aggiungere tutti gli anni relativi, per esempio se una versione è stata completata nel 2010 e un'altra nel 2012 allora andranno riportate entrambe le date (in questo caso " 2010, 2012 "), l'utilizzo di intervalli di date è permesso solo se in tutti gli anni che lo compongono sono state fatte delle nuove versioni e se se ne fa menzione esplicita nella documentazione (per esempio nel caso fosse " 2008, 2009, 2010, 2012 " si potrebbe scrivere " 2008-2010, 2012 "). Bisogna poi accertarsi di usare la parola "Copyright" come da convenzioni internazionali, e non una sua traduzione, mentre non è obbligatorio includere il simbolo (C). Nel caso sia stato usato anche del codice di un altro programma coperto dalla medesima licenza, si dovrà provvedere ad includere pure le sue note di copyright all'interno del file. Subito dopo la notifica di copyright bisogna mettere la dichiarazione di consenso alla copia e i termini per l'utilizzo. Un esempio del risultato finale è il seguente:

```
Copyright (C) <anno> <autore>
This file is part of <nome programma>
<nome programma> is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
any later version.

<nome programma> is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program. If not, see <http://www.gnu.org/licenses/>.

<eventuali informazioni di contatto come ad esempio un'email o un indirizzo
fisico di posta>
```

3.2 Preparare un pacchetto per la distribuzione con licenza GPLv343

```
<codice presente in quel file sorgente.>
```

Nel caso il programma richieda interazione col Terminale, è consigliato fargli stampare una breve notifica durante il suo avvio.

Un esempio può essere:

```
<nome programma> Copyright (C) <anno> <autore>
This program comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type 'show c' for details.
```

In questo caso specifico per esempio bisognerebbe implementare anche le istruzioni 'show w' e 'show c' ovviamente.

Nel caso invece si abbia un'interfaccia grafica si può utilizzare una finestra di dialogo "Informazioni su" o in inglese "about".

3.2.1 Includere codice preso da un altro progetto senza averlo modificato

Nel caso non si siano fatte modifiche al codice sorgente, dopo essersi assicurati che la licenza applicata ad esso sia compatibile con quella del proprio programma, bisogna preservare tutte le notifiche di copyright originali. Se su ognuno dei file inclusi sono già presenti le notifiche, allora bisogna lasciarli intatti, mentre nel caso esse siano state allegate in un file a parte, dovranno essere copiate all'inizio di ognuno dei file di quel progetto che saranno utilizzati. Nel secondo caso bisogna assicurarsi di includere tutti i nomi dei detentori del copyright sul file, se non è possibile distinguere dalla lista quali siano, è meglio includere tutti i nominativi presenti, e eventuali copyright, permessi accordati e avvisi di esonero dalle responsabilità richiesti dalla licenza preesistente.

3.2.2 Includere codice preso da un altro progetto dopo averlo modificato

In questo caso ci si trova di fronte a due opzioni: La prima è di aggiungere le proprie modifiche a quel/quei file sotto i termini della stessa licenza con cui erano stati rilasciati originariamente, e di aggiungere la propria notifica di copyright tra quelle di coloro che hanno contribuito alla scrittura del codice presente in quel file, oltre che soddisfare eventuali altre richieste di quella licenza. In questo caso però bisogna assicurarsi che i cambiamenti apportati non cambino il file in maniera tale che se preso separatamente dal progetto per cui è stato modificato esso venga comunque “invaso” dalla licenza GPL (per esempio interagendo con codice coperto dalla licenza GPL in una maniera tale che questa si “attacchi” anche ad esso). Questa opzione è solitamente utilizzata nel caso si voglia facilitare la collaborazione con il progetto da cui si è acquisito il codice originale, ma può creare il problema appena citato, quindi se si vuole mettere in atto questa opzione è meglio rivolgersi prima a qualcuno esperto in materia. La seconda opzione è quella di aggiungere le modifiche sotto i termini della licenza GPL, e quindi si dovrà aggiungere la propria notifica del copyright e quella riferita alla licenza all’inizio del file, assicurandosi di preservare anche le notifiche originali del file pur mantenendo i due gruppi distinti e ben identificabili. Per rendere più ordinate e chiare da individuare le varie notifiche relative alle parti coperte dalle licenze differenti è meglio non mischiarle tra loro, e fare in modo che le notifiche delle licenze più restrittive vengano aggiunte man mano all’inizio, mentre quelle delle licenze più permissive dovranno essere messe dopo le più restrittive. Un esempio può essere il seguente:

Notifica presente nel file prima di averlo modificato

```
\*
* Copyright (c) ANNO, NOME1
* Copyright (c) ANNO, NOME2
*
```

```
* <permessi e regole imposti dalla licenza>  
*  
* <avvisi di esonero dalle responsabilità>  
*/
```

Notifiche presenti dopo avervi aggiunto le proprie modifiche sotto licenza GPL

```
\*  
*Copyright (c) ANNO, NOME di chi ha modificato il file  
*  
*<permessi e regole imposti dalla licenza GPL>  
*  
*<avvisi di esonero dalle responsabilità>  
*  
*This file incorporates work covered by the following copyright and  
* permission notice:  
*****  
* Copyright (c) ANNO, NOME1  
* Copyright (c) ANNO, NOME2  
*  
* <permessi e regole imposti dalla licenza>  
*  
* <avvisi di esonero dalle responsabilità>  
*/
```

3.3 Licenziare XCMoel sotto GPLv3

In questa sezione applicheremo dal punto di vista pratico le informazioni che sono state illustrate precedentemente, per la verifica e l'ipotetica preparazione di XCMoel per il rilascio sotto licenza GPLv3.

Per prima cosa si è provveduto ad analizzare il contenuto della cartella principale del programma, in modo da individuare le varie componenti che lo compongono.

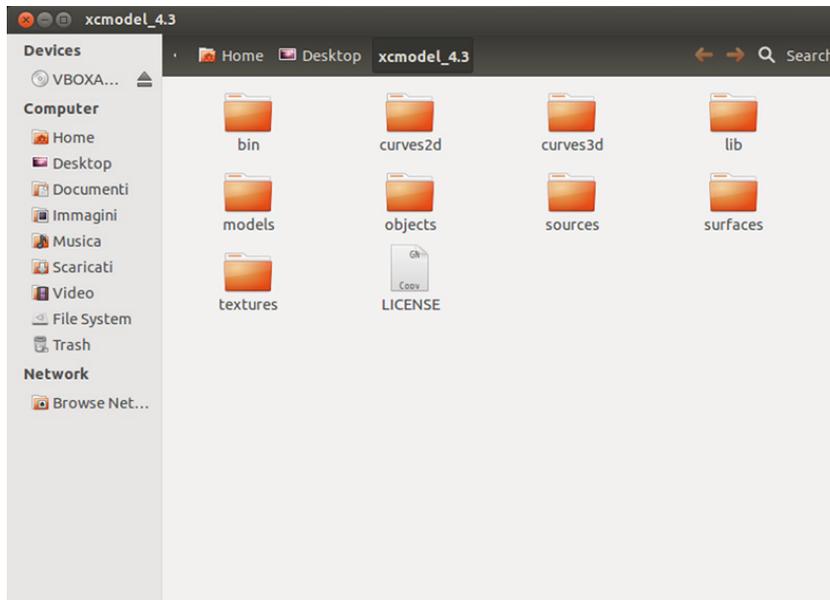


Figura 3.11: Contenuto cartella XCMoDel

Nella cartella chiamata “bin” sono presenti tre file che verranno usati da XCsurf per creare le finestre, e dopo aver compilato il relativo codice sorgente, in questa cartella saranno presenti anche gli eseguibili dei vari componenti che costituiscono il programma.

Sono poi presenti altre cartelle che contengono vari esempi di cosa può essere costruito utilizzando i vari pacchetti di XCMoDel. Questi esempi comprendono curve, superfici, textures, oggetti, e scene, in modo che l’utente possa iniziare a familiarizzare con i vari pacchetti sin da subito.

Infine c’è la cartella “sources” che contiene tutti i codici sorgente e gli script per compilarli.

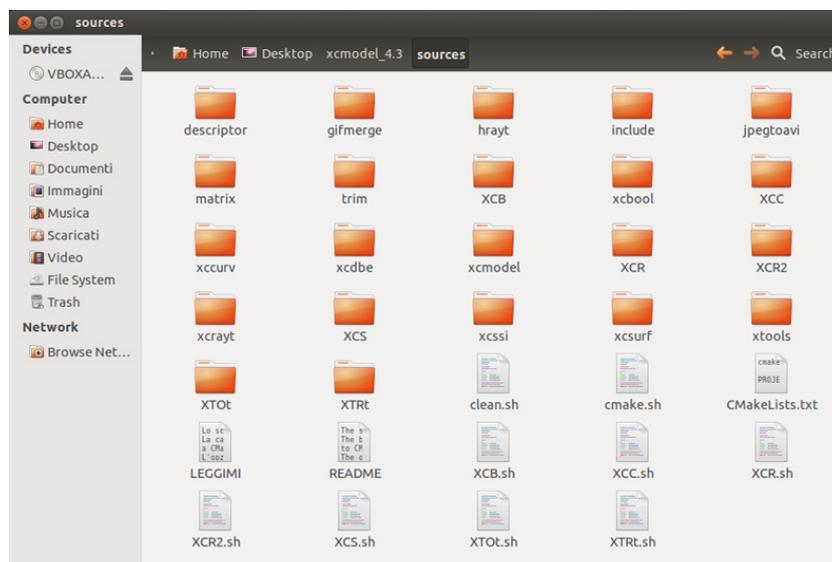


Figura 3.12: Contenuto sotto-cartella sources

Prima di procedere con l'applicazione della licenza scelta, in questo caso la GPL versione 3, bisogna assicurarsi che tutto sia conforme ai suoi termini. Esaminando i vari file del codice sorgente si può notare che in esso sono presenti le due cartelle “gifmerge” e “jpegtoavi”, che contengono i sorgenti di due programmi utilizzati da xcview per la conversione delle animazioni in formati più standard. Il primo programma serve a unire vari file immagine di formato gif in un unico file gif animato, in modo che esso contenga l'animazione, mentre il secondo provvede a creare un file avi da una serie di file in formato jpeg.

Procedendo ad esaminare prima il codice di “gifmerge” si può notare che esso è stato rilasciato sotto una licenza permissiva, però riscontriamo che nella notifica contenente le condizioni di distribuzione è contenuta la seguente frase:

- * This software may be freely copied, modified and redistributed*
- *without fee provided that this copyright notice is preserved*
- *intact on all copies and modified copies.*

Dato che l'imposizione dell'obbligo di poter distribuire gifmerge solo a condizione che questa distribuzione avvenga “without fee”, quindi gratuitamente, verrebbe considerata da GPLv3 come un'ulteriore restrizione imposta sul codice, questo software non potrebbe essere considerato compatibile.

Fortunatamente pur essendo il suo codice sorgente fornito insieme ai vari sorgenti dei pacchetti di XCMoel, essi non sono connessi direttamente fra loro, in quanto xcview comunica con “gifmerge” in una modalità che, secondo l'interpretazione ufficiale della licenza GPLv3, è considerata fra due programmi distinti⁶.

Ciò avviene in quanto “gifmerge” viene utilizzato da xcview facendo partire uno script bash (tramite l'istruzione in C “*execv*”), che avvia “gifmerge” con i parametri di ingresso necessari.

Se invece i due programmi fossero stati connessi tra loro in maniera più stretta, le opzioni sarebbero state le seguenti:

La prima è quella di riuscire a contattare i detentori del copyright su gifmerge e chiedere loro che ne rilascino una versione i cui termini siano compatibili con le condizioni imposte dalla licenza GPLv3.

La seconda opzione è invece quella di cercare del software che possa sostituire gifmerge e che abbia anche una licenza compatibile con la nostra.

Esaminando poi il codice di “jpegtoavi” notiamo che esso è licenziato sotto GPLv2 o versioni successive, quindi è utilizzabile all'interno del nostro programma licenziato con GPLv3, mentre se “jpegtoavi” fosse stato licenziato **solo** sotto GPLv2, non sarebbe stato compatibile, e quindi non sarebbe stato utilizzabile neppure esso. Non essendo stato modificato nessuno dei file sorgente di questo programma dobbiamo lasciare intatte le notifiche presenti su di esso.

A questo punto provvediamo ad inserire un file di testo “LICENSE”, contenente il testo completo della GPLv3, nella cartella principale (xcmo-

⁶<http://www.gnu.org/licenses/gpl-faq.en.html#MereAggregation>

del.4.3), per poi aggiungere le nostre notifiche all'inizio di tutti i file sorgente del nostro programma, aiutandosi con uno script bash scritto appositamente, dato il loro alto numero. Il testo inserito è il seguente:

```
/*
/*****
*Copyright (C) 2000,2002,2007,2013 Giulio Casciola
*This file is part of XModel
*XModel is free software: you can redistribute it and/or modify
*it under the terms of the GNU General Public License as published by
*the Free Software Foundation, either version 3 of the License, or
*(at your option) any later version.
*
*XModel is distributed in the hope that it will be useful,
*but WITHOUT ANY WARRANTY; without even the implied warranty of
*MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
*GNU General Public License for more details.
*
*You should have received a copy of the GNU General Public License
*along with this program. If not, see <http://www.gnu.org/licenses/>.
*
*
*For more informations on XModel visit:
* http://www.dm.unibo.it/~casciola/html/xcmodel.html
*
*****/

/* xcmodel.c */
#include <X11/Xlib.h>
#include <X11/Xutil.h>
```

Figura 3.13: Notifica inserita nei file sorgente di XModel

Infine si è provveduto ad inserire la notifica anche all'interno dell'interfaccia grafica, come da consiglio della GPLv3 stessa, e ad allegare una traduzione in inglese del file di testo contenente le istruzioni per compilare il programma o i suoi vari pacchetti.

A questo punto si può procedere a preparare il tutto per la distribuzione.

3.4 Diffondere il proprio programma Open Source

Nel caso si volesse facilitare la diffusione del proprio software open source, esistono varie possibilità di azione. Qualora si possedesse già un sito web, si potrebbe voler aumentare il numero di visite senza dover spendere soldi per pubblicizzarlo, in questo caso si può ricorrere all'utilizzo di alcuni

accorgimenti, come l'assicurarsi che il sito sia facile da usare e che le informazioni che esso fornisce siano facilmente accessibili, l'iscrizione del sito in varie web directory⁷, specialmente quelle che elencano siti dello stesso genere del proprio, fare in modo che nel testo del sito siano contenute le parole chiave più ricercate nei motori di ricerca riguardanti quell'argomento e in generale l'ottimizzazione del sito tramite l'utilizzo delle tecniche SEO⁸.

Nel caso non si possieda un sito, o comunque non ci si volesse limitare solo ad esso per diffondere la propria applicazione, si può anche scegliere di avvalersi degli strumenti offerti da siti specializzati esterni. Alcuni esempi dei più diffusi sono i seguenti:

Free Software Directory

È un progetto della Free Software Foundation avviato nel 1999, che cataloga il software libero in grado di funzionare su sistemi operativi liberi. Esso permette, previa registrazione, di pubblicizzare il proprio software gratuitamente e di fornire informazioni che possono essere utili per chi sta cercando un certo tipo di programma libero. Alcuni esempi di informazioni che possono essere fornite sono ad esempio: una breve descrizione del programma, un link alla pagina principale del progetto, alcune parole chiave che insieme alle categorie del software aiutino chi sta cercando un programma di quel tipo a trovarlo più facilmente, un link ai sorgenti, le varie licenze sotto cui è stato rilasciato, modalità per contattare l'autore e varie altre che possano essere ritenute utili. Una volta che si è inserito o aggiornato il proprio progetto e tutte le informazioni ad esso relative, esse non appariranno automaticamente nella parte pubblica del sito, ma saranno poste in attesa di approvazione da parte di uno degli amministratori, che provvederà a verificare che il programma e la licenza ad esso applicata siano compatibili con le linee guida del progetto. Al contrario dei siti che verranno elencati successivamente, esso non permette il caricamento dei sorgenti, ma può essere uno strumento utile,

⁷elenco di siti web suddivisi per categoria in maniera gerarchica

⁸Search Engine Optimization

da affiancare al proprio sito, per la pubblicizzazione e la diffusione del proprio software libero.

Sourceforge È una piattaforma fondata nel 1999, che permette agli sviluppatori di gestire e diffondere il proprio software open source gratuitamente. Fornisce vari servizi agli sviluppatori, tra cui alcuni esempi sono: un repository in cui caricare il proprio codice sorgente, un sistema per la gestione dei problemi che può essere utilizzato per ricevere notifica dei problemi riscontrati ad esempio durante l'esecuzione dagli utilizzatori del proprio programma, la possibilità di avere un forum di discussione moderabile dagli amministratori del progetto, una wiki per il progetto, la possibilità di visualizzare svariate statistiche riguardo i download e la possibilità da parte degli utenti di recensire e raccomandare i vari progetti. Per immettere o creare il proprio programma su Sourceforge bisogna prima di tutto aver registrato un account personale, dopodiché si può procedere con l'inserimento dei dati del progetto e la scelta di quali servizi usufruire (tenendo in considerazione che si potranno aggiungere o togliere servizi dal progetto che si sta amministrando anche successivamente). Se si è un admin del progetto si può decidere di aggiungere ad esso altri utenti sia come admin sia come sviluppatori, in modo che essi possano aiutare nello sviluppo. Tra i siti che sono qui presentati questo è l'unico che al momento permette ancora l'hosting dei binari del proprio progetto.

GitHub Fondato nel 2008, da allora è cresciuto rapidamente fino a diventare uno dei principali siti che offrono questo genere di servizio. Esso offre la possibilità di essere utilizzato sia come repository privato, che come repository di un progetto open source e, mentre nel primo caso è richiesto il pagamento di una quota mensile, nel secondo caso (cioè se il codice sorgente è pubblico) il servizio è fornito gratuitamente. Alcuni servizi che sono disponibili dopo essersi registrati come "free account" sono la possibilità di creare una wiki del progetto, la presenza di un tracker per poter gestire e tener traccia dei problemi riscontrati, la possibilità di analizzare e modificare

il codice contenuto nei sorgenti già caricati oltre alla possibilità di visualizzare alcune informazioni relative al progetto come ad esempio il numero di modifiche fatte al codice in un certo periodo.

3.5 Il dual licensing (o multi licensing)

Un modello di distribuzione del software molto diffuso è quello che sfrutta il dual licensing, in cui il programma è licenziato sotto 2 o più licenze differenti, di cui solitamente almeno una è open source. Esso permette di utilizzare la versione con la licenza più adatta alla specifica necessità ed è usato principalmente per due scopi principali, cioè il permettere la compatibilità con il maggior numero di licenze possibili, oppure come metodo di sfruttamento commerciale.

3.5.1 Dual licensing per la compatibilità con più licenze

In questo caso solitamente vengono rilasciate varie versioni del programma, ognuna delle quali licenziata sotto una licenza per il software libero o open source differente. Per esempio se un programma è stato rilasciato sotto GPLv3, nel caso qualcuno volesse usare parte di esso per un altro programma coperto da una licenza differente (sia che essa sia proprietaria che nel caso essa sia libera e/o open source), esso non potrebbe a causa della “viralità” di questa specifica licenza. In questo caso il problema può essere facilmente risolto nel caso il proprietario del copyright, su propria iniziativa o su richiesta altrui, rilasci i sorgenti del software anche sotto una licenza meno restrittiva e più adatta alle specifiche necessità insorte. Una delle licenze che il proprietario potrebbe scegliere come licenza alternativa è la LGPLv3, che permette l'utilizzo del codice a cui essa è stata applicata anche all'interno di codice coperto da licenze diverse, ovviamente a condizione che i vincoli da essa imposti vengano rispettati. Bisogna ricordarsi che se si ha intenzio-

ne di rilasciare una versione del software con una licenza diversa da quella originale, bisogna prima assicurarsi di avere il permesso di tutti i detentori del copyright, per cui se si è rilasciato del codice sotto una licenza, e questo codice è poi stato modificato e rilasciato da un'altra persona, si potrà rilasciare questa nuova versione sotto una licenza differente, solo con il permesso di colui che lo ha modificato, in caso contrario si potrà rilasciare sotto la nuova licenza solo la versione originale senza le modifiche (o in caso esso sia stato modificato da più persone, solo la versione con le modifiche di coloro che hanno dato l'autorizzazione).

3.5.2 Dual licensing come metodo per lo sfruttamento commerciale

L'altro metodo di dual licensing (anche se non tutti concordano nell'includerlo in questa categoria) è solitamente sfruttato dalle imprese per avere un mezzo di guadagno pur rilasciando il software anche sotto licenza open source.

La sua forma più comune consiste nel rilasciare gratuitamente la versione coperta da una licenza open source molto restrittiva, come può essere ad esempio la GPLv3, che permetta la modifica e la distribuzione solo se viene fornito anche tutto il codice sorgente più eventuali altre restrizioni specifiche della licenza. Queste restrizioni renderebbero molto problematico utilizzare il codice di quel software come base per la creazione di programmi commerciali, e quindi a questo punto entra in gioco il dual licensing con cui l'impresa che possiede quel codice può fornire ai richiedenti la versione del codice con applicata una licenza che ne permetta lo sfruttamento commerciale anche in ambito proprietario, in cambio del pagamento di un compenso. Questa seconda licenza può spesso essere di tipo commerciale, e ciò implica che la maggior parte del codice deve essere sviluppato dall'impresa, in quanto non sempre un collaboratore esterno per la versione open è disposto a permettere che il suo contributo sia sfruttato commercialmente. Per risolvere questo problema alcune imprese richiedono ai collaboratori esterni l'accettazione di

un contratto di licenza (Contributor License Agreement⁹) prima di accettare le loro modifiche nella versione ufficiale. Questo contratto concede al proprietario il diritto irrevocabile di redistribuire quella modifica come parte del programma anche sotto altre licenze.

Un altro problema in cui può incorrere un'impresa che attua questa strategia di guadagno, è quello del forking del progetto da parte di qualche competitore o uno o più utenti insoddisfatti di come il software viene gestito dal proprietario. Ciò può causare la nascita di un diretto concorrente, che oltretutto fa uso (almeno inizialmente) del proprio codice rilasciato sotto licenza open source, e quindi con buone basi per divenire un concorrente temibile. Il modo migliore per prevenire ciò è quello di mantenere buoni rapporti con la comunità a cui ci si rivolge, ascoltandone le richieste e facendo in modo che non ritenga più conveniente appropriarsi essa stessa del controllo sul software.

3.6 Modelli di guadagno

Come già accennato in precedenza, nonostante molti credano il contrario, né il software libero né quello open source devono essere per forza distribuiti gratuitamente. Infatti pur essendo questo tipo di software difficilmente commerciabile secondo il classico modello del software proprietario, esso permette comunque di essere utilizzato come mezzo di guadagno sia dal singolo che dall'impresa.

Oltre al metodo del dual licensing già spiegato in precedenza, ve ne sono molti altri, alcuni dei quali ritenuti interessanti anche per progetti di dimensioni limitate e che verranno quindi mostrati in questa sezione.

- **Scissione tra software open source e prodotto commerciale:**

Un metodo di guadagno è quello di rilasciare con una licenza permissiva che permetta di mischiare software libero o open source con quello proprietario. Consiste nel rilasciare, sotto licenza open source di tipo

⁹<http://oss-watch.ac.uk/resources/cla>

non copyleft, una versione semplificata del software con solo le funzioni base, mentre la versione commerciale conterrà tutte le funzionalità e sarà ottenibile solo con una licenza proprietaria. Questo metodo ha però il problema del dover bilanciare le due parti, in modo che gli utenti a cui il software è indirizzato siano invogliati a pagare per ottenere la versione completa, piuttosto che limitarsi ad usare la versione FOSS, e contemporaneamente si deve sperare anche che altri sviluppatori non decidano di ampliare la versione open source con funzionalità equivalenti a quelle presenti nella versione proprietaria.



Figura 3.14: Scissione tra software open source e prodotto commerciale

- **Vendita dei servizi da parte degli specialisti del prodotto:** Questo modello di guadagno è solitamente usato per i programmi completamente open source, e consiste nel fornire gratuitamente il software, ma di far pagare eventuali mezzi di formazione e consulenza forniti, sfruttando il fatto che solitamente coloro che hanno prodotto quel de-

terminato software vengono percepiti come le persone che lo conoscono meglio che quindi essere la fonte principale a cui rivolgersi in caso di difficoltà. Il suo lato negativo è però che, qualora qualcun altro acquisisse le conoscenze necessarie, esso potrebbe mettersi in concorrenza diretta nella fornitura di quei servizi.

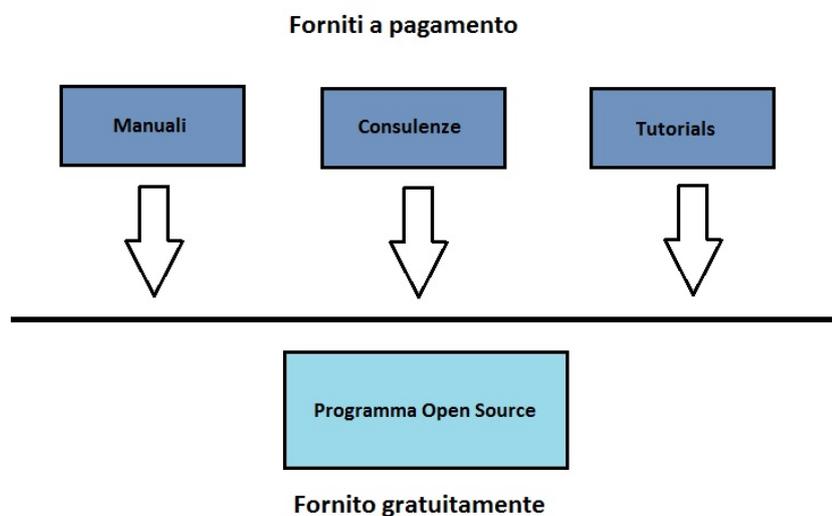


Figura 3.15: Prodotto fornito gratuitamente ma vendita dei servizi

- **Donazioni:** Con questo metodo si distribuisce il software gratuitamente, e se l'utente lo ha trovato utile può decidere di donare liberamente, a seconda delle sue possibilità, per incentivare il proseguimento dello sviluppo.
- **Open Sourcing ritardato:** Consiste nel rilasciare la versione più recente del software inizialmente sotto licenza proprietaria e a pagamento, ovviamente rispettando i termini della licenza applicata ad eventuali parti open source, che non deve essere quindi di tipo copyleft forte, e dopo un certo periodo di tempo, rilasciare quella versione completamente sotto licenza open source.

3.7 Come comportarsi se qualcuno infrange la licenza

Se ci si accorge che qualcuno usa un programma licenziato sotto GPL senza rispettarne le condizioni, bisogna avvisare il proprietario (o i proprietari) del copyright, in quanto essi sono gli unici che possono far valere i propri diritti su di esso in forma legale.

Se invece si viene a conoscenza di una qualche infrazione della licenza di un programma su cui si detiene personalmente il copyright, allora si deve notificare l'infrazione a colui che la sta commettendo, tramite mezzi ragionevoli. A questo punto il trasgressore ha 30 giorni dalla ricezione della comunicazione per sanare la violazione; se questo avviene ed era la sua prima infrazione su uno dei nostri lavori allora il problema è considerato risolto ed egli può tornare a farne uso (ovviamente questa volta deve rispettare i termini della licenza). Passato quel periodo, a meno che non decidiamo di terminare definitivamente i suoi diritti sul nostro software, esso ha comunque la possibilità di cessare ogni violazione delle condizioni della licenza, in tal caso i suoi diritti saranno ripristinati temporaneamente. Se invece il trasgressore si è accorto dell'infrazione prima di noi e ha provveduto a risolverla da almeno 60 giorni, allora non c'è bisogno del nostro intervento, in quanto i suoi diritti di utilizzare il nostro software rilasciato sotto licenza GPLv3 sono stati ripristinati automaticamente.

Nel caso invece esso non dia segno di volersi mettere in regola, una volta passati i 30 giorni dalla comunicazione di infrazione, possiamo comunicare la terminazione permanente di tutti i suoi diritti concessi dalla licenza sul nostro lavoro, a causa della sua contravvenzione. A questo punto le opzioni disponibili dipendono da svariati fattori, tra cui il proprio stato di residenza, lo stato di residenza del trasgressore, e le varie leggi relative al copyright in essi promulgate. Per decidere quale sia il miglior percorso per gestire la violazione il metodo migliore è quello di rivolgersi a siti specializzati nel-

l'argomento, tra cui i principali sono "Software Freedom Law Center"¹⁰ e "gpl-violations.org" per chiedere informazioni riguardo al proprio specifico caso.

Un altro sito che può rivelarsi molto utile è "Plagiarism Today"¹¹, in cui oltre alle varie risorse e consigli forniti riguardo ai problemi col copyright, vengono forniti anche alcuni template di lettere.

In generale comunque le opzioni principali sono: il procedimento legale, che però ovviamente richiede una certa quantità di risorse, monetarie e non (e il risultato può variare a seconda dello stato in cui viene attuato), e la richiesta di rimozione del materiale soggetto a copyright agli host in quanto secondo le leggi DMCA¹² (per gli U.S.A.) e EU Electronic Commerce Directive¹³ (per l'Unione Europea negli stati in cui è stata ratificata) un host online non è responsabile per l'attività illegale immessa nel suo sistema da un utente solo finché l'host non ne viene messo a conoscenza, una volta ottenuta questa conoscenza l'host deve rimuovere o impedire l'accesso al materiale in questione. La seconda opzione permette di richiedere per esempio a motori di ricerca come Google e Yahoo di rimuovere gli indirizzi incriminati dalle pagine indicizzabili, riducendo così la possibilità di distribuzione del software incriminato.

¹⁰<http://softwarefreedom.org/>

¹¹<http://www.plagiarismtoday.com/>

¹²<http://www.copyright.gov/legislation/hr2281.pdf>

¹³<http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32000L0031:IT:HTML>

Conclusioni

Come si è visto, il mondo delle licenze libere e open source è molto vasto e complicato, soprattutto per chi non è un esperto in argomenti come il diritto d'autore o le licenze. Oltretutto, mentre è abbastanza facile trovare informazioni generali riguardo alle varie licenze, trovare informazioni specifiche riguardo il loro utilizzo soprattutto nella parte pratica è invece piuttosto difficile. Per esempio trovare informazioni su come ci si può tutelare nel caso ci si accorga che il proprio programma sia sfruttato da altre persone che non rispettano i termini della licenza, ha richiesto ricerche molto approfondite del web, prima di trovare ciò che si cercava.

Nonostante tutto risulta comunque evidente che lo sviluppo e l'utilizzo di software open source si trova decisamente in un periodo di espansione, come dimostrato anche dal fatto che molti programmi di uso più o meno comune rientrano in questa categoria. Ciò ha permesso la nascita di valide alternative, anche in ambiti in cui ormai regnava il monopolio di uno o più programmi, contribuendo quindi a mantenere quelle applicazioni dinamiche e aggiornate, rispetto alle necessità degli utenti, invece di imporre le funzioni e utilità più convenienti per i produttori.

Rispetto alla classica metodologia di sviluppo del software in cui esso viene realizzato da un gruppo chiuso composto da uno o più sviluppatori con ruoli più o meno definiti, i programmi open source hanno la possibilità di essere sviluppati in maniera più dinamica, in quanto ogni collaboratore può decidere di modificare le parti del codice in cui ritiene sia più necessario, ed eventualmente può anche aggiungere funzionalità che ritiene utili. Ciò

comporta che un programma possa rimanere attivo e dinamico anche dopo che gli sviluppatori originari hanno smesso di supportarlo, dato che se esso è ritenuto utile per la comunità, troverà facilmente delle persone disposte a proseguire il suo aggiornamento. In definitiva, al giorno d'oggi è molto utile conoscere come funziona il mondo del software libero e open source, sia per chi volesse contribuirvi direttamente, ma anche per chi volesse solo sapere del perché di certe scelte riguardanti programmi che rientrano in quella categoria.

Bibliografia

[1] Free Software Foundation

<http://www.fsf.org/>

[2] Open Source Initiative

<http://opensource.org/>

[3] Testo GPLv3, LGPLv3, AGPLv3

<http://gplv3.fsf.org/>

[4] XCModel

<http://www.dm.unibo.it/~casciola/html/xcmodel.html>

[5] Software Freedom Law Center

<http://www.softwarefreedom.org>

[6] Free Software Directory

http://directory.fsf.org/wiki/Main_Page

[7] Sourceforge

<http://sourceforge.net>

[8] GitHub

<https://github.com>

-
- [9] Andrew Updegrave *A Concise Introduction to Free and Open Source Software*, Standards Today, August September 2009 Vol VIII No 5
http://www.consortiuminfo.org/bulletins/pdf/aug09/aug09_feature.pdf
- [10] Rachna Kapur, Mario Briggs, Tapas Saha, Ulisses Costa, Pedro Carvalho, Raul F. Chong, Peter Kohlmann *Getting started with open source development*, IBM Corporation, 2010
- [11] Simone Aliprandi *Compendio di libertà informatica e cultura open*, PrimaOra , 2006
<http://www.copyleft-italia.it/libri.html>
- [12] Elena Grandi *Introduzione al mondo del Software Libero e dell'Open Source*, 2004
<http://www.pluto.it/ildp/doc-it/intro.html>
- [13] G. Casciola *xcmodel: a system to model and render NURBS curves and surfaces, User's Guide*, Version 1.0, (2000)
<http://www.dm.unibo.it/~casciola/html/xcmodel.html>