9

LICENTIES, AUTEURSRECHTEN EN PATENTEN

De licentie die u kiest, heeft waarschijnlijk geen grote invloed op de aanvaarding van uw project, zolang het een open source-licentie betreft. Gebruikers kiezen software normaal gesproken om hun kwaliteit en gebruiksmogelijkheden, niet om de details van de licentie. Desalniettemin moet u wel het één en ander weten over opensoftwarelicenties, enerzijds om u ervan te verzekeren dat de licentie van het project past bij het doel van het project en anderzijds om over de keuze van de licentie te kunnen discussiëren. Houd wel in de gaten dat ik geen jurist ben en dat niets in dit hoofdstuk geïnterpreteerd mag worden als formeel juridisch advies. Daarvoor zult u een jurist moeten inschakelen, tenzij u er zelf één bent.

9.1 TERMINOLOGIE

In elke discussie over open source-licenties wordt direct duidelijk dat er veel verschillende benamingen lijken te zijn voor hetzelfde: *vrije software*, *open source*, *FOSS*, *F/OSS* en *FLOSS*. Laten we deze en wat andere termen eerst aan een nader onderzoek onderwerpen.

Free software

Software kan vrij gedeeld en aangepast worden, inclusief de broncode ervan. Deze term werd voor het eerst gebruikt door Richard Stallman. Hij is degene die het gecodificeerd heeft in de GNU General Public License (GPL) en is ook de oprichter van de Free Software Foundation (http://www.fsf.org/). Deze stichting is opgericht om het concept free software te promoten.

Hoewel ‘free software’ bijna de lading van ‘open source’-software dekt, geven de FSF en vele anderen de voorkeur aan de eerste term, omdat deze de nadruk legt op het idee van vrijheid en het concept van het vrij te verspreiden software, hoofdzakelijk als een sociale beweging in plaats van een technische. De FSF geeft toe dat de term dubbelzinnig is, het kan ‘vrij’ zijn in de zin van ‘kosteloos’, in plaats van ‘vrij’ als in ‘vrijheid’, maar ze vinden dat dit toch de beste term is. De alternatieven hebben ook hun dubbelzinnigheden. (In dit hele boek door is ‘vrij’ gebruikt in de zin van ‘vrijheid’ en niet van ‘kosteloos’.)

Open source-software

Open source-software is een andere naam voor vrije software. Deze andere naam staat echter voor een belangrijk filosofisch verschil. 'Open source' is geïntroduceerd door het Open Source Initiative (http://www.opensource.org/) als goed alternatief voor 'vrije software'. En om te laten zien dat dergelijke software een aantrekkelijke keus is voor bedrijven wordt het gepresenteerd als een ontwikkelingsmethodologie in plaats van een politieke beweging. Ze hebben wellicht ook een ander stigma willen overwinnen, namelijk dat alles wat 'vrij' is, haast wel van slechte kwaliteit moet zijn.

Omdat iedere licentie die gratis is ook open source is en vice versa (een enkele uitzondering daargelaten) hebben mensen de neiging één term te kiezen en daar ook aan vast te houden. Over het algemeen is het zo dat degenen die 'free software' prefereren waarschijnlijk een meer filosofische of morele kijk op de zaak hebben, terwijl degenen die 'open source' prefereren, het of niet zien als een kwestie van vrijheid, of niet de behoefte voelen om ermee te koop te lopen. Zie het gedeelte ''Vrij' versus 'open source'' in Hoofdstuk 1, *Introductie* voor een meer gedetailleerde uiteenzetting over deze tweedeling.

De Free Software Foundation geeft een goede (beslist niet objectieve, maar wel genuanceerde en behoorlijk eerlijke) uitleg over de twee termen op http://www.fsf.org/licensing/essays/free-software-for-freedom.html. De uitleg van het Open Source Initiative staat verspreid over twee sites: http://www.opensource.org/advocacy/case_for_hackers.php#marketing en http://www.opensource.org/advocacy/free-notfree.php.

FOSS, *F/OSS*, *FLOSS*

Als er twee zijn van iets, dan worden het er snel drie, en dat is precies wat er aan het gebeuren is met de termen voor vrije software. De academische wereld, in een poging precisie en alomvattendheid te verkrijgen in plaats van elegantie, lijkt gekozen te hebben voor FOSS, of soms F/OSS, wat staat voor 'Free / Open Source Software'. Een andere variant die aan populariteit wint, is FLOSS, wat staat voor 'Free / Libre Open Source Software' (*libre* is bekend in vele talen en niet onderhevig aan de dubbelzinnigheid van 'gratis/free'. Zie http://en.wikipedia.org/wiki/FLOSS voor meer hierover).

Al deze termen betekenen in essentie hetzelfde: software die door iedereen aangepast kan worden en doorgegeven, soms, maar niet altijd, met de vereiste dat afgeleide werken vrij verspreid kunnen worden maar onder de dezelfde voorwaarden.

DFSG-compliant

Compliant met the Debian Free Software-richtlijnen (http://www.debian.org/social_contract#guidelines). Dit is een veelgebruikte test om te kijken of een bepaalde licentie werkelijk open source (vrij, *libre* enz.) is. De missie van het Debian-project is om een geheel gratis besturingssysteem op te zetten, en wel zo dat iemand die het installeert zich nooit hoeft af te vragen of hij het mag aanpassen of het gedeeltelijk of helemaal mag doorgeven. De richtlijnen van Debian Free Software zijn de vereisten waaraan een licentie van een softwarepakket moet voldoen om in Debian opgenomen te worden. Omdat het Debian-project erg veel tijd besteed heeft aan de opzet van de test, zijn de richtlijnen waarmee ze kwamen zeer robuust gebleken (zie http://en.wikipedia.org/wiki/DFSG). Voor zover ik weet is er geen bezwaar tegen deze richtlijnen geuit, niet door de Free Software Foundation en ook niet door het Open Source Initiative. Als u weet dat een gegeven licentie conform DFSG is, dan weet u dat het alle belangrijke vrijheden garandeert (zoals forkability, zelfs tegen de wens van de oorspronkelijke auteur in), wat een vereiste is om de dynamiek van een open source-project in stand te houden. Alle besproken licenties in dit hoofdstuk zijn conform DFSG.

OSI-goedgekeurd

Goedgekeurd door het Open Source Initiative. Dit is een andere veelgebruikte manier om te zien of een licentie alle benodigde vrijheden biedt. De definitie van open source-software van OSI is gebaseerd op de Debian Free Software-richtlijnen, en licenties die voldoen aan de ene definitie doen dat bijna ook altijd aan de andere. Er zijn de laatste jaren enkele uitzonderingen geweest, maar deze betroffen alleen nichelicenties en zijn niet relevant hier. In tegenstelling tot het Debian-project houdt het OSI een lijst bij met alle licenties die ze ooit goedgekeurd hebben op http://www.opensource.org/licenses/, zodat 'OSI-approved' een ondubbelzinnige status is. Een licentie komt voor of komt niet voor op de lijst.

De Free Software Foundation houdt ook een lijst bij met licenties op http://www.fsf.org/licensing/licenses/license-list.html. De FSF categoriseert licenties niet alleen naar of ze vrij zijn maar ook of ze compatibel zijn met de GNU General Public-licentie. De compatibiliteit van een GPL is een belangrijk onderwerp. Dit wordt behandeld in het gedeelte 'De GPL en compatibiliteit van licenties' verderop in dit hoofdstuk.

Propriëtair, *closed-source*

Dit is het tegenovergestelde van 'vrije software' of 'open source'. Het betekent software verspreidt onder de traditionele, op royalty's gebaseerde licenties, waarbij gebruikers betalen per kopie, of onder andere licenties die voldoende beperkend zijn zodat de open source-dynamiek niet in werking kan treden. Zelfs software die gratis wordt verspreidt kan propriëtair zijn, als de licentie erop geen verdere verspreiding of aanpassing toestaat.

Over het algemeen worden 'propriëtair' en 'closed-source' gebruikt als synoniemen. Maar 'closed-source' impliceert ook nog dat zelfs de broncode niet gezien kan worden. En omdat de broncode bij de meeste propriëtaire software niet zichtbaar is, komt dit dus meestal op hetzelfde neer. In een enkel geval brengt iemand echter software uit onder een licentie die toestaat om de broncode te bekijken. Verwarrend is wel dat ze dit soms 'open source' of 'bijna open source' noemen, maar dit is misleidend. De *zichtbaarheid* van de broncode is niet van belang, de belangrijke vraag is wat iemand ermee mag *doen*. Daaruit volgt dat het verschil tussen propriëtaire software en closed-source bijna geheel irrelevant is en deze twee gezien kunnen worden als synoniemen.

Soms wordt het woord *commercieel* gebruikt als een synoniem voor 'propriëtair,' maar feitelijk zijn deze twee niet hetzelfde. Vrije software kan commerciële software zijn. Want uiteindelijk kan vrije software verkocht worden zolang de kopers er maar niet van weerhouden worden kopieën weg te geven. Het kan ook op andere manieren commercieel gebruikt worden, bijvoorbeeld door de verkoop van ondersteuning, service of certificering. Er bestaan bedrijven waarin vele miljoenen dollars omgaan en die gegrondvest zijn op vrije software. Vrije software is dus duidelijk intrinsiek anticommercieel noch antibedrijfsmatig. Aan de andere kant is de aard ervan wel antipropriëtair en dit is de essentie van het verschil met traditionele modellen waarbij *licenties per kopie* worden verkocht.

Publiek domein

Het feit dat software geen auteursrechten heeft, betekent dat er niemand is die het recht bezit om kopiëren van het werk te beperken. Bij het publieke domein horen is niet hetzelfde als geen auteur hebben. Alles heeft een auteur en al kiest de auteur of de auteurs ervoor het werk in het publieke domein te plaatsen, verandert dat niets aan het feit dat ze het geschreven hebben.

Wanneer een werk in het publieke domein is opgenomen, kan materiaal daarvan ingelijfd worden in werk waar wel auteursrechten op zit. Daarna valt *die kopie* van het materiaal onder dezelfde auteursrechten als het gehele werk. Maar dit heeft geen effect op de verkrijgbaarheid van het originele werk, wat beschikbaar blijft in het publieke domein. Dus iets vrijgeven in het publieke domein is technisch gezien een manier om het werk 'gratis of vrij' te maken, in overeenstemming met de richtlijnen van de meeste voor 'vrije software' certificerende organisaties. Hoe dan ook, er zijn meestal goede redenen om een licentie aan te vragen in plaats van het werk vrij te geven in het publieke domein. Zelfs bij vrije software kunnen bepaalde restricties zinvol zijn, niet alleen voor degene die het auteursrecht bezit maar ook voor de ontvanger van het werk. Dit wordt duidelijk in het volgende gedeelte.

Auteursrechtenvrij (copyleft)

Een licentie die auteursrechtelijke wetten toepast om het tegenovergestelde te bereiken van het traditionele auteursrecht. Al naar gelang aan wie je het vraagt, betekent het licenties die de vrijheden toestaan die hier ter discussie staan, of licenties die niet alleen die vrijheden toestaan maar ook afdwingen, door te bedingen dat deze vrijheden onlosmakelijk verbonden zijn met het werk. De Free Software Foundation gebruikt louter de tweede, meer specifieke, definitie. Bij de andere partijen is het een beetje een gok. Vele gebruiken de term op de manier waarop de FSF dat doet, maar andere (waaronder ook mensen die voor de reguliere media schrijven) neigen naar de eerste definitie. Het is niet voor iedereen die deze term bezigt duidelijk dat er een onderscheid gemaakt moet worden.

Het algemeen aanvaarde voorbeeld van de preciezere en striktere definitie is de GNU General Public-licentie, die bepaalt dat elk afgeleid werk ook onder de GPL-licenties moet vallen. Zie het gedeelte 'De GPL en licentiecompatibiliteit' verderop in dit hoofdstuk voor meer hierover.

9.2 ASPECTEN VAN LICENTIES

Hoewel er veel verschillende vrijesoftwarelicenties zijn, zeggen ze op de belangrijkste punten allemaal hetzelfde: dat iedereen de code aan mag passen, dat iedereen deze mag verspreiden, zowel in de originele als in de aangepaste vorm, en dat de houders van de auteursrechten geen garanties geven (wettelijke aansprakelijkheid vermijden is van uitzonderlijk belang gezien het feit dat mensen een aangepaste versie kunnen gebruiken zonder dit zelf te weten). De verschillen tussen licenties zijn in het kort samen te vatten tot enkele veel voorkomende kwesties:

Compatibiliteit met propriëtaire licenties

Sommige vrije licenties staan toe dat de betreffende code wordt gebruikt in propriëtaire programma's. Dit heeft geen gevolgen voor de licentievoorwaarden van het propriëtaire programma: het blijft net zo propriëtair, alleen bevat wat code uit een niet propriëtaire bron. De Apache-licentie, de X Consortium-licentie, de BSD-licenties en de MIT-licenties zijn allemaal voorbeelden van licenties die compatibel zijn met propriëtaire software.

Compatibiliteit met andere vrije licenties

De meeste vrije licenties zijn compatibel met elkaar. Dat wil zeggen dat code onder de ene licentie, gecombineerd kan worden met code onder een andere licentie, en dat het resultaat onder één van beide licenties verspreid kan worden zonder de voorwaarden van de ander te schenden. De grote uitzondering hierop is the GNU General Public-licentie, die vereist dat elk werk dat gebruik maakt van code met een GPL verder gedistribueerd moet worden onder de GPL, zonder toevoeging van extra restricties buiten die van de GPL.

De GPL is compatibel met enkele vrije licenties, maar niet met alle. Hierop wordt verder ingegaan in het gedeelte 'De GPL en compatibiliteit van licenties' verderop in dit hoofdstuk.

Handhaving van credits

Sommige vrije licenties vereisen dat elk gebruik van de bedoelde code vergezeld gaat van een mededeling, waarvan de plaatsing en presentatie meestal wordt aangegeven, om op deze manier credit te geven aan de auteurs van de code. Deze licenties zijn vaak nog steeds compatibel met propriëtaire software: ze vereisen niet per definitie dat het afgeleide werk vrij is, alleen maar dat de vrije code wordt voorzien van een credit.

Bescherming van handelsmerk

Dit is een variant op credithandhaving. Licenties die handelsmerken beschermen, geven aan dat de naam van de originele software (of van degenen die de auteursrechten hebben, of het instituut enz.) *niet* gebruikt mag worden bij afgeleide werken zonder voorafgaande schriftelijke toestemming. Terwijl crediting van een licentie vereist dat een bepaalde naam vermeld wordt, vereist een handelsmerkbescherming dat het niet gebruikt mag worden. Ze drukken echter beide dezelfde wens uit, namelijk dat de reputatie van de originele code bewaard en doorgegeven wordt, en niet bezoedeld wordt door de associatie met een ander product.

Bescherming van 'artistieke integriteit'

Sommige licenties (de Artistic-licentie, gebruikt voor de meest populaire implementatie van de programmeertaal Perl en Donald Knuths TeX-licentie bijvoorbeeld) vereisen dat de modificatie en verspreiding op een dusdanige manier gedaan worden dat er een duidelijk onderscheid gemaakt wordt tussen de onaangetaste originele versie van de code en de modificaties. In principe staan ze dezelfde vrijheden toe als andere licenties, maar leggen ze bepaalde eisen op die het makkelijk maken om de integriteit van het origineel te verifiëren. Deze licenties zijn buiten het specifieke programma waar ze voor gemaakt zijn nauwelijks gebruikt en zullen niet verder besproken worden in dit hoofdstuk. Ze worden hier alleen genoemd voor de volledigheid.

Bij de meeste van deze bepalingen sluit de ene de andere niet uit en sommige licenties bevatten verscheidene. De rode draad hier is dat ze eisen stellen aan de ontvanger in ruil voor het recht van de gebruiker op gebruik en/of verspreiding van de code. Bij sommige projecten willen ze bijvoorbeeld dat de naam en de reputatie doorgegeven wordt met de code en vinden ze dat de extra moeite van een credit- of handelsmerkclausule waard. De zwaarte van de verplichtingen echter, kan een reden zijn voor sommige gebruikers om voor een pakket met een minder dwingende licentie te kiezen.

9.3 DE GPL EN LICENTIECOMPATIBILITEIT

Verreweg de meest duidelijke scheidslijn tussen licenties is die tussen wel propriëtair-compatibele en niet-propriëtair-compatibele licenties. Dat is dus de scheidslijn tussen de GNU General Public-licentie en alle anderen. Omdat het primaire doel van de GPL-auteurs de promotie van vrije software is, hebben ze de licentie opzettelijk zo gemaakt dat het onmogelijk is om de code met GPL te gebruiken in propriëtaire programma's. Specifieke GPL-vereisten (zie http://www.fsf.org/licensing/licenses/gpl.html voor de volledige tekst) zijn deze twee:

1. Elk afgeleid werk, dat wil zeggen, elk werk dat een niet onbelangrijk deel code bevat met GPL, moet verspreid worden onder de GPL.

2. Er mogen geen restricties toegevoegd worden, niet aan het origineel en niet aan het afgeleide werk. (De exacte omschrijving is: "You may not impose any further restrictions on the recipients' exercise of the rights granted herein." NL: "U mag geen verdere beperkingen opleggen inzake de rechten van de ontvangers dan degenen die hierin zijn opgenomen.")

Met deze voorwaarden is de GPL erin geslaagd het open maken van software aanstekelijk te maken. Als van een programma de auteursrechten eenmaal vastgelegd zijn onder de GPL, worden de voorwaarden voor verspreiding als een *virus* verspreid. Dat wil zeggen dat ze overgaan op alles waarin de code opgenomen wordt, hetgeen het onmogelijk maakt om code met GPL te gebruiken in programma's met closed-source. Dezelfde clausules maken de GPL echter soms niet compatibel met sommige andere vrije licenties. Gewoonlijk gebeurt dit doordat andere licenties een eis opleggen (bijvoorbeeld een creditclausule die vereist dat de originele auteurs op een bepaalde wijze genoemd worden) die niet compatibel is met de vereiste van de GPL. "U mag geen verdere beperkingen opleggen ..." . Vanuit het oogpunt van de Free Software Foundation zijn de consequenties van deze bijkomende eisen wenselijk, of op zijn minst niet te negatief. De GPL houdt niet alleen uw software vrij, maar maakt uw software een doeltreffende vertegenwoordiger van vrije software en kan zo ook andere software pushen vrijheden toe te staan.

De vraag of dit een goede manier is om vrije software te promoten, blijft één van de meest hardnekkige heilige oorlogen op internet (zie het gedeelte 'Heilige oorlogen voorkomen' in Hoofdstuk 6, *Communicatie*)), hier zullen we dit niet verder bespreken. Wat belangrijk is voor ons doel is dat compatibiliteit met de GPL een belangrijke kwestie is wanneer u een licentie gaat kiezen. De GPL is verreweg de meest populaire open source-licentie. Op http://freshmeat.net/stats/#license scoort het 68% en de eerstvolgende in de rij van gekozen licenties staat op 6%. Als u wilt dat uw code vrij gebruikt kan worden met de code met GPL (en er is erg veel code met GPL in omloop) dan moet u een licentie kiezen die compatibel is met de GPL. De meeste open source-licenties die compatibel zijn met de GPL zijn ook compatibel met propriëtaire licenties, Oftewel, code onder een dergelijke licentie kan gebruikt worden in een programma met GPL en het kan gebruikt worden in een propriëtair programma. Het *resultaten* van combinaties van deze twee zijn uiteraard niet compatibel met elkaar, omdat de één onder de GPL zal vallen en de ander onder een closed-source-licentie. Maar die zorg treft alleen de afgeleide werken en niet de code die u oorspronkelijk verspreidde.

Gelukkig houdt de Free Software Foundation een lijst bij waarin alle licenties zijn opgenomen die compatibel zijn met de GPL en welke dat niet zijn, op http://www.gnu.org/licenses/license-list.html. Alle besproken licenties in dit hoofdstuk staan op die lijst, aan de ene kant of aan de andere.

9.4 EEN LICENTIE KIEZEN

Wanneer u een licentie voor uw project wilt kiezen, kies dan als het mogelijk is een bestaande licentie in plaats van een nieuwe te maken. Er zijn twee redenen waarom een bestaande licentie beter is:

- Bekendheid. Als u één van de drie of vier meest populaire licenties kiest dan hebben mensen niet het gevoel dat ze eerst al het juridische jargon moeten lezen om uw code te kunnen gebruiken, omdat ze dit lang geleden al gedaan hebben voor die licentie.
- Kwaliteit. Tenzij u een team juristen tot uw beschikking hebt, is het onwaarschijnlijk dat u met een legale en solide licentie op de proppen kunt komen. De hier genoemde licenties zijn het resultaat van veel overdenkingen en ervaring. Tenzij uw project uiterst ongewone behoeften heeft, is het onwaarschijnlijk dat u het er beter van afbrengt.

Raadpleeg het gedeelte 'Hoe u een licentie toepast op uw software' in Hoofdstuk 2,

Het begin om te weten te komen hoe u één van deze licenties kunt aanvragen voor uw project.

De MIT / X Window System-licentie
Als uw doel is om uw code toegankelijk te maken voor het grootst mogelijke aantal ontwikkelaars en afgeleide werken, en u hebt er geen bezwaar tegen dat uw code gebruikt wordt in propriëtaire programma's, kies dan de MIT / X Window System-licentie (deze is zo genoemd omdat het de licentie is waaronder the Massachusetts Institute of Technology de originele X Window System-code heeft uitgebracht). De primaire boodschap van deze licentie is "U bent vrij om deze code te gebruiken zoals u wilt." Hij is compatibel met de GNU GPL, hij is kort, simpel en makkelijk te begrijpen:

Copyright (c) <year> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(Overgenomen van http://www.opensource.org/licenses/mit-license.php.)

De GNU General Public-licentie
Als u niet wilt dat de code van uw project gebruikt wordt in propriëtaire programma's of als het u niets kan schelen of het wel of niet gebruikt kan worden in propriëtaire programma's, kies dan de GNU General Public-licentie (http://www.fsf.org/licensing/licenses/gpl.html). De GPL is vandaag de dag waarschijnlijk de meest gebruikte licentie voor open source-software ter wereld. Directe herkenbaarheid is een van GPL's grootste voordelen.

Als u een codebibliotheek maakt die voornamelijk is bedoeld om gebruikt te worden als een gedeelte van andere programma's, overweeg dan zorgvuldig of de restricties die opgelegd zijn door de GPL stroken met het doel van uw project. In sommige gevallen (bijvoorbeeld als u probeert een concurrerende propriëtaire bibliotheek uit het zadel te wippen) kan het wat strategisch inzicht vereisen om voor uw code een licentie te vinden die gebruikt kan worden in combinatie met propriëtaire programma's, ook al zou u dit in eerste instantie niet gewild hebben. De Free Software Foundation heeft zelfs een alternatief gemaakt voor de GPL voor dergelijke situaties: de *GNU Library GPL*, die later de nieuwe naam *GNU Lesser GPL* kreeg (de meeste mensen gebruiken het acroniem *LGPL*). De LGPL heeft minder strenge restricties dan de GPL en kan makkelijker worden gecombineerd met niet-vrije code. Hoe dan ook, het blijft een beetje complex en het vraagt wat tijd om het te kunnen begrijpen. Dus als u de GPL niet gaat gebruiken, raad ik u aan de MIT/X-style-licentie te nemen.

Is de GPL gratis of niet gratis?
Een consequentie van de keuze voor de GPL is de kans (die kans is klein, maar niet oneindig klein) dat uw project verwikkeld raakt in een conflict over de vraag of de GPL daadwerkelijk 'vrij' is, gegeven het feit dat deze enkele restricties oplegt over wat u mag doen met de code, namelijk de restrictie dat de code niet meer verspreid mag worden onder een andere licentie. Voor sommige mensen betekent het bestaan van deze restrictie dat de GPL 'minder vrij' is dan de meer tolerante licenties als de MIT/X-licentie. Waar deze discussie meestal in uitmondt, is dat 'vrijer' beter is dan 'minder vrij' (wie is er per slot van rekening geen voorstander van vrijheid?) en dat deze licenties dus wel beter moeten zijn dan de GPL.

Deze discussie is een van de populaire heilige oorlogen (zie het gedeelte 'Heilige oorlogen voorkomen' in Hoofdstuk 6*, Communicatie*). Laat u zich niet tot dergelijke discussie verleiden, in ieder geval niet op de projectforums. Probeer niet te bewijzen dat de GPL minder vrij, even vrij of vrijer is dan andere licenties. Benadruk in plaats daarvan de specifieke redenen waarom er binnen uw project gekozen is voor de GPL. Als de herkenbaarheid van de licentie de reden was, zeg dat dan. Als de handhaving van een vrije licentie over de afgeleide werken ook een reden was, meld dat dan ook, maar weiger deel te nemen aan de discussie over of het deze code meer of minder 'vrij' maakt. Vrijheid is een ingewikkelde materie en er is weinig reden om over de terminologie te praten als dit gebruikt gaat worden als afleiding van het onderwerp waar het in wezen om draait.

Dit is een boek en geen thread op een mailinglijst, maar ik moet toegeven dat ik het argument 'GPL is niet vrij' nooit begrepen heb. De enige restrictie die de GPL oplegt, is degene die mensen er van weerhoudt *verdere* restricties op te leggen. Om te zeggen dat dit resulteert in minder vrijheid doet me voorgekomen als zeggen dat slavernij verbieden vrijheid reduceert, omdat het sommige mensen ervan zal weerhouden slaven te hebben.

(O ja, en als u zich dan toch bij deze discussie hebt laten betrekken, drijf de boel dan niet op de spits door opruiende analogieën aan te halen.)

En de BSD-licentie?
Een behoorlijk deel van alle open source-software is verspreid onder een *BSD-licentie* (of soms een *BSD-achtige licentie*). De originele BSD-licentie is gebruikt voor de Berkeley Software Distribution, waaronder de Universiteit van Californië belangrijke gedeeltes van de Unix-implementatie heeft uitgebracht. Deze licentie (de exacte

tekst kunt u vinden in sectie 2.2.2 op http://www.xfree86.org/3.3.6/COPYRIGHT2.html#6) is van gelijke strekking als de MIT/X-licentie, op één clausule na:

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, Lawrence Berkeley Laboratory.

De aanwezigheid van die clausule maakte niet alleen de oorspronkelijke BSD-licentie incompatibel met GPL, maar creëerde ook een gevaarlijk precedent. Toen andere organisaties gelijke advertentieclausules in *hun* vrije software gingen zetten (en hun eigen naam lieten toevoegen in plaats van 'the University of California, Lawrence Berkeley Laboratory') werd het voor softwareverspreiders steeds moeilijker te bepalen wat ze moesten vermelden. Gelukkig werden veel van de projecten die deze licentie gebruikten zich bewust van het probleem en hebben ze simpelweg de advertentieclausule geschrapt. In 1999 heeft zelfs de Universiteit van Californië dit gedaan.

Het resultaat is een herziene versie van de BSD-licentie, die gewoon de oorspronkelijke BSD-licentie is maar dan zonder de advertentieclausule. Deze achtergrond maakt de term 'BSD-licentie' echter een beetje dubbelzinnig. Verwijst hij naar de oorspronkelijke of de herziene versie? Dit is waarom de MIT/X-licentie is te prefereren. Deze is in essentie gelijkwaardig, maar heeft niet te lijden van enige dubbelzinnigheid. Er is misschien één reden om de voorkeur te geven aan de herziene versie van de BSD-licentie boven de MIT/X-licentie, en dat is deze clausule uit BSD-licentie:

Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

Het is onduidelijk of zonder een dergelijke clausule een ontvanger van de software het recht zou hebben om de naam van de licentiehouder te gebruiken. Deze clausule echter, neemt iedere mogelijke twijfel weg. Voor organisaties die bezorgd zijn over de controle over hun handelsmerk, valt de herziene versie van de BSD-licentie wellicht net te prefereren boven die van de MIT/X. Algemeen impliceert een liberaal auteursrecht gewoonlijk niet dat de ontvangers uw handelsmerken mogen gebruiken of verzwakken. De wet op de auteursrechten en de wet op de handelsmerken zijn twee zeer verschillende dingen.

Als u de herziene versie van de BSD-licentie wilt gebruiken, is er een sjabloon beschikbaar op http://www.opensource.org/licenses/bsd-license.php.

9.5 TOEKENNEN VAN AUTEURSRECHT EN EIGENDOM

Er zijn drie manieren om om te gaan met het eigendom van auteursrechten voor vrije code en documentatie die ontwikkeld zijn door meerdere mensen. De eerste is om de kwestie van het auteursrecht geheel te negeren (ik raad dit niet aan).

De tweede is om een *contributor license agreement* (*CLA*) van iedere persoon die binnen het project werkzaam is te verkrijgen, waarin aan het project uitdrukkelijk het recht wordt toegekend om de bijdragen van die persoon te gebruiken. Dit is gewoonlijk voldoende voor de meeste projecten en het prettige is dat in sommige rechtsgebieden een CLA per e-mail verzonden mag worden. De derde manier is om daadwerkelijke eigendomsoverdracht te krijgen van auteursrechten van mensen die bijdragen, zodat het project (d.w.z. een legale eenheid, meestal een non-profit-organisatie) de eigenaar is van alle auteursrechten. Dit een legale en waterdichte manier, maar het geeft ook de meeste rompslomp voor degenen die bijdragen aan het project. Er zijn dan ook maar enkele projecten die er op staan dat het op deze manier gebeurt.

Er moet opgemerkt worden dat zelfs bij gecentraliseerd geregelde auteursrechten de code vrij blijft[31], omdat open source-licenties de auteursrechthebbende niet het recht geven om met terugwerkende kracht zich alle kopieën van de code toe te eigenen. Dus zelfs als het project, als legale eenheid, plotseling van mening zou veranderen en zou beginnen alle code onder een licentie met veel restricties uit te brengen dan zou dit geen problemen geven voor de publieke gemeenschap. De andere ontwikkelaars zouden simpelweg een fork starten gebaseerd op de laatste vrije kopie en verdergaan alsof er niets aan de hand is. Omdat men weet dat ze dit kunnen doen werken de meeste mensen die bijdragen mee als hun wordt gevraagd om een CLA of een toekenning van de auteurrechten te ondertekenen.

Niets doen

Bij de meeste projecten worden van de contribuanten geen CLA's of auteursrechtoverdracht gevraagd. In plaats daarvan accepteren ze code telkens wanneer het redelijkerwijze duidelijk is dat het de bedoeling was van de contribuant om de code bij het project in te lijven.

Onder normale omstandigheden is dit in orde. Zo nu en dan kan er echter iemand beslissen om een rechtzaak aan te spannen over schending van auteursrechten en beweren dat hij de ware eigenaar is van de code in kwestie en dat hij het project nooit toestemming gegeven heeft om de code te verspreiden onder een open source-licentie. De SCO Group deed bijvoorbeeld iets dergelijks bij het Linux project. Zie http://en.wikipedia.org/wiki/SCO-Linux_controversies voor details. Wanneer dit gebeurt, beschikt het project niet over documentatie om aan te kunnen tonen dat de contribuant het recht om de code te gebruiken heeft overgedragen, wat het verdedigen in een rechtzaak moeilijker kan maken.

Contributor-licentieovereenkomsten

CLA's bieden waarschijnlijk de beste balans tussen veiligheid en gemak. Een CLA is meestal een elektronisch formulier dat een ontwikkelaar invult en opstuurt naar het project. In vele rechtsgebieden is een verklaring per e-mail genoeg. Soms is een gewaarborgde digitale ondertekening vereist. Raadpleeg een jurist om na te gaan welke optie het beste is voor uw project.

De meeste projecten gebruiken twee enigszins verschillende CLA's, één voor individuen en één voor bedrijven. Voor beide types geldt echter dat de kern hetzelfde is:

de contribuant kent het project *"...eeuwigdurende, wereldwijde, niets uitsluitende, kosteloze, royaltyvrije en onherroepelijke auteursrechtlicentie om te reproduceren, afgeleide werken te maken, publiekelijk te presenteren, publiekelijk uit te voeren, sublicenties op aan te vragen en bijdragen en daarvan afgeleide werken te distribueren"* toe. Ook hier zou u een jurist in de arm moeten nemen om de CLA goed te keuren, maar als u al deze toevoegingen erin zet, zit u waarschijnlijk goed.

Wanneer u uw contribuanten vraagt een CLA te ondertekenen, wees dan zeker dat u duidelijk maakt dat u *niet* om het feitelijke auteursrecht vraagt. Veel CLA's beginnen met een opmerking voor de lezer:

Dit is alleen een licentieovereenkomst; deze draagt geen auteursrechten over en verandert niets aan uw rechten om uw bijdragen te gebruiken voor welk ander doel dan ook.

Hier zijn enkele voorbeelden:

- CLA's voor individuele contribuanten:
  - http://apache.org/licenses/icla.txt
  - http://code.google.com/legal/individual-cla-v1.0.html
- CLA's voor bedrijven:
  - http://apache.org/licenses/cla-corporate.txt
  - http://code.google.com/legal/corporate-cla-v1.0.html

Overdracht van auteursrecht

Overdracht van auteursrecht betekent dat de contribuant het auteursrecht van zijn bijdragen toekent aan het project. Dit moet schriftelijk worden gedaan en per fax of per post naar het project worden gezonden.

Binnen sommige projecten staat men op volledige overdracht omdat het nuttig kan zijn om het auteursrecht van de gehele code onder te brengen bij één rechtspersoon voor het geval de voorwaarden van de open source-licentie afgedwongen moeten worden in een rechtzaak. Als er niet één rechtspersoon is die het recht heeft om dit te doen dan moeten alle ontwikkelaars meewerken. Sommigen hebben wellicht geen tijd echter, of zijn zelfs niet bereikbaar op het moment dat de zaak voorkomt.

Niet alle organisaties zijn even streng op het gebied van het inzamelen van auteursrechtoverdracht. Sommige vragen alleen om een informele verklaring van een codecontribuant op een publieke mailinglijst, vaak in de trant van "Ik verklaar hierbij het auteursrecht van deze code toe te kennen aan het project waarbij de code onder dezelfde licentie valt als al de andere code." Minstens één advocaat die ik hierover gesproken heb heeft gezegd dat dit voldoende moet zijn, vermoedelijk omdat het in een context staat waar het overdragen van auteursrecht normaal en verwacht is, en omdat het op een betrouwbare manier laat zien dat het project moeite doet om de werkelijke bedoelingen van de ontwikkelaar te achterhalen. De Free Software Foundation houdt echter vast aan het andere uiterste. Ze eisen van de contribuanten dat ze een formele schriftelijke verklaring ondertekenen waarop staat dat ze het auteursrecht overdragen, soms voor een enkele bijdrage en soms voor lopende of toekomstige bijdragen. Als de ontwikkelaar een werkbetrekking heeft vraagt de FSF de werkgever ook te tekenen.

Deze paranoia van de FSF is te begrijpen. Als iemand de rechten van de GPL schendt door iets van hun software in te lijven in een propriëtair programma, dan zal de FSF dat aanvechten in een rechtzaak, en ze willen een zo waterdicht mogelijk auteursrecht wanneer dit gebeurt. Sinds de FSF auteursrechthouder is van veel populaire software denken ze dat er een reële kans bestaat dat dit gebeurt. Of uw organisatie even nauwgezet te werk moet gaan, is iets dat alleen u kunt beslissen in overleg met uw advocaten. Tenzij er een specifieke reden is waarom uw project volledig auteursrecht nodig heeft, kunt u normaal gesproken gewoon de CLA's gebruiken; dat is het makkelijkst voor iedereen.

9.6 TWEEVOUDIGE LICENTIEPROGRAMMA'S

Sommige projecten proberen aan inkomsten te komen door een tweevoudig licentieprogramma te gebruiken, waarin afgeleide werken met auteursrecht de auteursrechthouder voor het gebruik van de code kunnen betalen, maar waarbij de code nog wel vrij te gebruiken blijft voor open source-projecten. Dit werkt doorgaans natuurlijk beter bij codebibliotheken dan losstaande toepassingen. De exacte voorwaarden verschillen van geval tot geval. Vaak is de licentie voor het vrije gedeelte de GNU GPL, omdat deze anderen toch al verbiedt de beschermde code in hun Auteursrechtelijk beschermde product te gebruiken zonder toestemming van de auteursrechthouder, maar het kan ook een op maat gemaakte licentie zijn met hetzelfde effect. Een voorbeeld van de eerstgenoemde is de MySQL-licentie, zoals te zien op http://www.mysql.com/company/legal/licensing/, een voorbeeld van de tweede is Sleepycat Softwares licentiestrategie, te zien op http://www.oracle.com/technology/software/products/berkeley-db/htdocs/licensing.html.

U zou zich af kunnen vragen hoe een houder van auteursrechten een licentie kan aanbieden voor een verplicht bedrag als de GNU GPL-licentie vereist dat de code verkrijgbaar moet zijn onder veel minder beperkende voorwaarden? Het antwoord hierop is dat de auteursrechthouder de GPL voorwaarden oplegt aan alle anderen; de eigenaar is daarom vrij om te besluiten dat deze voorwaarden *niet* voor hemzelf gelden. U kunt zich voorstellen dat de auteursrechthouder een oneindig aantal kopieën van de software opgeslagen heeft. Iedere keer dat hij er eentje pakt om de wereld in te sturen kan hij beslissen wat voor licentie hij daarop toepast: GPL, auteursrechten of iets anders. Zijn recht om dit te doen is niet gebonden aan de GPL of aan welke open source-licentie dan ook. Het is simpelweg een recht dat hij heeft op basis van de wet op auteursrechten.

De aantrekkelijkheid van tweevoudige licenties is dat het een manier verschaft aan open source-softwareprojecten zichzelf van een betrouwbare inkomstenstroom te voorzien. Helaas botst dit soms met de normale dynamiek van open source-projecten. Het probleem is dat iedere vrijwilliger die een code bijdraagt dit nu doet aan twee verschillende entiteiten: de vrije versie van de code en de versie met auteursrechten. Terwijl de contribuant zich prettig kan voelen met zijn bijdrage aan de vrije

versie, omdat dat nu eenmaal de norm is binnen open source-projecten, kan hij een onprettig gevoel overhouden aan het feit dat hij ook bijdraagt aan iemands semi-propriëtaire inkomstenbron. Dit is niet erg elegant, wat wordt verergerd door het feit dat bij tweevoudige licenties de eigenaar daadwerkelijk formele en getekende auteursrechttoekenning moet gaan verzamelen van alle contribuanten, om op deze manier te voorkomen dat een ontevreden contribuant van code later een percentage gaat claimen van de royalty's. Dit proces van het verzamelen van de toekenningspapieren confronteert de contribuanten met het feit dat ze bezig zijn geld te verdienen voor iemand anders.

Niet alle vrijwilligers zullen zich hieraan storen. Uiteindelijk komen hun bijdragen ook terecht in de open source-versie, wat precies in hun belang is. Desalniettemin, tweevoudige licenties zijn een voorbeeld van een speciaal recht dat de auteursrechthouder zich toekent, dat de anderen binnen het project niet hebben en daarom zeker tot spanning kan leiden, in ieder geval bij enkele vrijwilligers.

Wat er gebeurt in de praktijk is dat bedrijven die gebaseerd zijn op software met tweevoudige licenties niet daadwerkelijk een egalitaire ontwikkelaarsgemeenschap hebben. Ze krijgen kleinschalige bugfixes en patches van externe bronnen, maar uiteindelijk moeten ze het echte werk zien te klaren met interne middelen. Zack Urlocker, Vice President marketing bij MySQL, vertelde bijvoorbeeld dat de meeste bedrijven uiteindelijk toch de meest actieve vrijwilligers moesten inhuren. Dus hoewel het project zelf open source is met een licentie onder de GPL, wordt de ontwikkeling ervan in meer of mindere mate gecontroleerd door het bedrijf, ondanks het (zeer onwaarschijnlijke) feit dat iemand die ontevreden is met de manier waarop het bedrijf met de software omgaat een fork maakt van het project. Tot op welke hoogte dit invloed heeft op het beleid van een bedrijf weet ik niet, maar in ieder geval lijkt MySQL geen acceptatieproblemen te hebben, niet in de open source-wereld en ook niet daarbuiten.

9.7 PATENTEN

Softwarepatenten zijn momenteel een heet hangijzer binnen de vrije software, omdat ze de enige bedreiging zijn waartegen de vrije softwaregemeenschap zich niet kan verdedigen. Problemen met auteursrechten en handelsmerken zijn wel te omzeilen. Als het erop lijkt dat een gedeelte van uw code inbreuk maakt op het auteursrecht van een ander, dan kunt u dat gedeelte gewoon herschrijven. En als het blijkt dat iemand een handelsmerkrecht heeft op de naam van uw project dan moet u in het ergste geval de naam van het project veranderen. Alhoewel een naamsverandering tijdelijk ongemakken geeft, maakt het op de lange duur niet uit, omdat de code op zich nog steeds doet wat hij altijd deed.

Een patent is echter een allesomvattend verbod op het implementeren van een bepaald idee. Het maakt niet uit wie de code schrijft of welke programmeertaal gebruikt wordt. Op het moment dat iemand een open source-softwareproject ervan beschuldigt inbreuk te maken op een patent, dan moet het project of stoppen met het implementeren van die bepaalde functie of een dure en tijdrovende rechtszaak voeren. Omdat de aanstichters van dergelijke rechtszaken meestal bedrijven zijn met veel geld (dat zijn sowieso degenen die de middelen en de neiging hebben om patenten aan te vragen) kunnen de meeste open source-softwareprojecten zich de laatste mogelijkheid niet permitteren en moeten zich onmiddellijk gewonnen geven, zelfs als ze denken dat ze een goede kans maken dat het patent voor de rechter niet standhoudt. Om te voorkomen dat ze in een dergelijke situatie verzeild raken, zetten open source-softwareprojecten hun code steeds vaker defensief op, om op voorhand gepatenteerde algoritmes te vermijden ook al zij die de best bruikbare of de enige beschikbare oplossing voor een probleem in een programma.[32]

Samenvattingen en anekdotisch bewijs laten zien dat niet alleen de overgrote meerderheid van open source-programmeurs, maar een meerderheid van *alle* programmeurs, vindt dat patenten op software afgeschaft moeten worden.[33] Open source-programmeurs vinden dit vaak uitgesproken belangrijk en kunnen weigeren aan projecten mee te werken die te sterk verband houden met het verkrijgen of handhaven van softwarepatenten. Als uw organisatie softwarepatenten verzamelt, maak dan publiekelijk en onherroepelijk duidelijk dat deze patenten nooit zullen worden ingezet tegen open source-projecten en dat ze alleen gebruikt mogen worden als verdediging voor het geval een andere partij een rechtszaak wegens inbreuk begint tegen uw organisatie. Dit is niet alleen correct, het is ook goede reclame voor de open source-wereld.[34]

Jammer genoeg is het defensief verzamelen van patenten een rationele actie. Het huidige patenteersysteem, in ieder geval in de Verenigde Staten, lijkt wel op een wapenwedloop. Als uw mededingers veel patenten hebben verworven, dan is de beste verdediging om zelf een hoop patenten te verzamelen, zodat wanneer u een rechtszaak aangespannen krijgt wegens inbreuk op een patent u dit kunt beantwoorden met een vergelijkbare dreiging. Op dat moment gaan de beide partijen meestal om de tafel zitten om een afspraak over de licentie te maken zodat beide partijen niets hoeven te betalen, behalve hun advocaten natuurlijk.

De schade die open source-software ondervindt door softwarepatenten is hoe dan ook verraderlijker dan een direct gevaar voor de ontwikkeling van code. Softwarepatenten moedigen een sfeer aan van geheimzinnigheid rondom de ontwikkelaars van firmware, die terecht bezorgd zijn dat door het publiceren van details van hun interfaces ze technische hulp geven aan concurrenten die een manier zoeken om een rechtszaak aan te spannen wegens inbreuk op een patent. Dit is niet slechts een theoretisch gevaar. Het gebeurt klaarblijkelijk al heel lang in de videokaartensector bijvoorbeeld. Veel producenten van videokaarten zijn terughoudend met het vrijgeven van de gedetailleerde programmaspecificaties die nodig zijn om krachtige open source-drivers te produceren voor hun kaarten. Zo maken ze het onmogelijk voor vrij opererende systemen om deze kaarten voor hun volle capaciteit te ondersteunen. Waarom doen de producenten dit? Het lijkt onlogisch als ze softwareondersteuning *tegenwerken.* Uiteindelijk betekent compatibiliteit met meerdere besturingssystemen alleen maar meer verkoop van kaarten. Het blijkt echter dat, achter de deur van de ontwerpkamer, deze bedrijven allemaal elkaars patenten schenden, soms per ongeluk en soms willens en wetens. De patenten zijn zo onvoorspelbaar en potentieel zo breed dat geen enkele kaartproducent ooit zeker weet of hij goed

zit, zelfs niet na een patentenonderzoek. Daarom durven producenten hun volledige interfacespecificaties niet te publiceren omdat dat het voor de concurrenten veel gemakkelijker zou maken om uit te zoeken of er inbreuk op hun patenten gedaan gepleegd. (Uiteraard is de aard van deze situatie zo, dat u dit nergens zwart op wit en uit de eerste hand zult terugvinden; ik ben het te weten gekomen in een persoonlijk gesprek.)

Sommige vrijesoftwarelicenties hebben speciale clausules om softwarepatenten te bevechten of in ieder geval te ontmoedigen. De GNU GPL bijvoorbeeld bevat deze tekst:

7. If, as a consequence of a court judgment or allegation of patentinfringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence youmay not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

[...]

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

De Apachelicentie, versie 2.0 (http://www.apache.org/licenses/LICENSE-2.0) bevat ook antipatentvoorwaarden. Allereerst bepaalt de licentie dat iedereen die code verspreidt onder deze licentie impliciet een royaltyvrije licentie moet hebben voor eventuele patenten die op de code van toepassing zouden kunnen zijn. Ten tweede straft het iedereen op ingenieuze wijze af die een inbreuk claimt op het beschermde werk, door automatisch hun impliciete patentlicentie uit te schakelen op het moment dat ze een claim indienen:

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

Alhoewel het nuttig is, zowel juridisch als politiek gezien, om op deze manier een verdediging tegen patenten in te bouwen in open source-softwarelicenties, zullen deze stappen uiteindelijk niet genoeg zijn om het verkillende effect dat de dreiging van rechtszaken om patenten heeft op vrije software weg te nemen.

Alleen veranderingen aan de inhoud of de interpretatie van internationaal patentrecht kunnen dat doen.

Om meer te weten te komen over dit probleem en hoe het aangepakt kan worden, ga naar http://www.nosoftwarepatents.com/. Het volgende artikel op Wikipedia http://en.wikipedia.org/wiki/Software_patent biedt ook veel bruikbare informatie over softwarepatenten. Ik heb ook een blogpost geschreven met daarin de samenvatting van de argumenten tegen software patenten, op http://www.rants.org/2007/05/01/how-to-tell-that-software-patents-are-a-bad-idea/.

9.8 VERDERE BRONNEN

Dit hoofdstuk is alleen een introductie over open source-softwarelicenties. Ik hoop hoe dan ook dat het genoeg informatie bevat om u op gang te helpen met uw eigen open source-project. Ieder serieus onderzoek naar licentiekwesties zal meer opleveren dan dit boek aan informatie kan bieden. Hier volgt een lijst met meer bronnen over open source-licenties:

- *Understanding Open Source and Free Software Licensing* door Andrew M. St. Laurent. Gepubliceerd door O'Reilly Media, eerste editie August 2004, ISBN: 0-596-00581-4.

  Dit is een boek van gemiddelde lengte over open source-licenties in al hun complexiteit, inclusief veel onderwerpen die zijn weggelaten in dit hoofdstuk. Kijk op http://www.oreilly.com/catalog/osfreesoft/ voor details.

- *Make Your Open Source Software GPL-Compatible. Or Else.* Door David A. Wheeler, op http://www.dwheeler.com/essays/gpl-compatible.html.

  Dit is een goed geschreven en gedetailleerd artikel over waarom het belangrijk is om een GPL-compatibele licentie te nemen, ook al gebruikt u de GPL-licentie zelf niet. Het artikel behandelt ook veel andere licentiekwesties en bevat een groot aantal zeer goede links.

- *http://creativecommons.org/*
  Creative Commons is een organisatie die een stelsel van meer flexibele en

liberale auteursrechten promoot dan de huidige toepassingen van het auteursrecht. Ze bieden niet alleen licenties voor software, maar ook voor tekst, kunst en muziek, allemaal toegankelijk via een gebruiksvriendelijke licentiekiezer. Sommige licenties zijn copylefts, sommigen zijn niet-copyleft maar nog steeds gratis, andere zijn gewoonweg traditionele auteursrechten met wat minder strikte beperkingen. De website van Creative Commons geeft zeer duidelijke uitleg over waar het allemaal om gaat. Al ik een site zou moeten kiezen om de bredere filosofische implicaties van de vrijesoftwarebeweging te demonstreren, dan zou ik deze kiezen.

30| De overdracht van auteursrechten is op dit moment onderhevig aan nationale wetgeving en licenties ontworpen voor de Verenigde Staten kunnen ergens anders problemen geven (bijv. in Duitsland, waar het klaarblijkelijk niet mogelijk is auteursrechten over te dragen).

31| Ik zal vanaf nu het woord 'code' gebruiken om te verwijzen naar zowel code als documentatie.

32| Sun Microsystems en IBM hebben in ieder geval een gebaar gemaakt naar aanleiding van dit probleem door een groot aantal softwarepatenten vrij te geven (respectievelijk 1600 en 500) voor gebruik door de open source-gemeenschap. Ik ben geen jurist en kan daarom de werkelijke bruikbaarheid niet beoordelen van deze concessies, maar zelfs als het allemaal belangrijke patenten zijn, en de voorwaarden maken ze daadwerkelijk vrij voor gebruik door ieder open source-project, dan nog is het een druppel op een gloeiende plaat.

33| Zie http://lpf.ai.mit.edu/Whatsnew/survey.html voor een samenvatting.

34| RedHat heeft bijvoorbeeld toegezegd dat open source-projecten veilig zijn voor hun patenten, zie http://www.redhat.com/legal/patent_policy.html.