

Regulatory and Proprietary Aspects Related to the Use of *FLOSS* for Collaborative Knowledge Construction in an Academic Environment

Tarsila de Conti
Universidade de São Paulo (USP)
tarsilaconti@yahoo.com.br

Mardel de Conti
Universidade de São Paulo (USP)
mbdconti@usp.br

Abstract

This paper analyzes some regulatory and proprietary aspects related to the use of *FLOSS* (Free / Libre / Open Source *Software*, or variants) for collaborative knowledge construction in an academic environment, be them more intensively directed to research, to technical application or to learning activities. The use, improvement and development of *software* under different types of licenses are analyzed and discussed in respect to applicable regulation, constraints to be followed, recognition of intellectual property, motivation for adoption, and other matters. Collaborative knowledge construction by multiple users, openly accessed or with stratified log-in restrictions, are analyzed and discussed in respect to the same aspects as above. Finally, a case study is analyzed, where both *FLOSS* and collaborative knowledge construction philosophies are integrated aiming research, application and learning of numerical methods in engineering.

1. Introdução

Este *paper* aborda o regime de propriedade intelectual aplicável a *softwares* desenvolvidos sob um modelo de trabalho colaborativo em ambiente acadêmico. O foco mais específico desta análise é auxiliar a implantação e o uso de um ambiente computacional para pesquisa, aplicações e aprendizagem de métodos numéricos em engenharia. Para tanto, serão apresentados os principais conceitos jurídicos que regulam a propriedade sobre os *softwares* e os principais tipos de licença aplicáveis. Finalmente, será realizado um breve estudo de caso em que se ponderam os modelos de licença mais adequados às situações vislumbradas pelo projeto.

2. Licenças de *Software* - *Softwares* com Direitos Autorais e *Softwares* em Domínio Público; Licença Proprietária e Licença “*FLOSS*”; Licença “*FLOSS*” Permissiva e Licença “*FLOSS*” *Copyleft*

Segundo a convenção de Berna, os *softwares* são regidos pelos mesmos direitos conferidos às obras literárias, isto é, o regime de direitos autorais. Estes direitos compõem um conjunto de

prerrogativas exclusivas do autor sobre o *software*, instituindo um verdadeiro regime de propriedade (neste caso, propriedade intelectual).

Os direitos autorais nascem automaticamente a partir da criação do *software*, independem de qualquer formalidade ou registro, e perduram durante cinquenta anos a partir da data de sua publicação. Deste modo, a existência de direitos autorais sobre um *software* corresponde à condição jurídica natural, é o padrão legal, e independe qualquer postura ativa do autor. Já a abdicação ou a cessão destes direitos corresponde à situação jurídica excepcional, e depende ou de uma ação deliberada do autor ou da transcorrência do prazo legal.

Cada tipo de objeto enseja diferentes modos de manifestação dos direitos de propriedade. Com relação aos direitos autorais sobre *softwares*, incidem as prerrogativas exclusivas de uso, modificação e distribuição do produto. Em outras palavras, todo uso do *software* é legalmente reservado ao autor; trata-se portanto de uma abordagem jurídica negativa, em que se exclui todo uso por parte de terceiros.

Esta abordagem é especialmente relevante em se tratando de bens intangíveis, não exclusivos por natureza. A imaterialidade do bem (como é o caso da informação e do conhecimento) garante sua reprodutibilidade ao infinito, de modo que todos podem possuí-lo ao mesmo tempo e por inteiro. É uma situação bastante diferente da dos bens materiais, em que o uso por mais de um sujeito só pode ocorrer de modo parcial ou alternado. O sistema de propriedade intelectual garante a exclusividade sobre bens imateriais de modo a contornar sua natureza ilimitada.

Cada um dos direitos garantidos pelo sistema de propriedade intelectual pode ser transferido de modo independente dos demais, seja de modo gratuito ou mediante a instituição de alguma obrigação ou pagamento. A transferência destes direitos ocorre através de instrumentos jurídicos particulares, em que se abdicam determinadas prerrogativas de direito autoral em face do uso por terceiros. Assim, o que se percebe é que contratos e licenças são essenciais ao sistema de propriedade intelectual; são meios de compartilhamento de direitos de propriedade de modo controlado.

No que diz respeito aos *softwares*, a transferência de propriedade intelectual ocorre, no mais das vezes, por via de licenças.

As licenças são acordos de com termos de natureza unilateral que adquirem legitimidade jurídica a partir de um ato de aceitação da outra parte. Assim, as licenças assemelham-se a contratos de adesão entre o distribuidor original do *software* e o usuário-final, de modo que ao usuário-final não cabe a discussão dos termos, mas apenas sua aceitação ou rejeição.

Via de regra, as licenças de softwares apresentam-se sob a forma de documentos eletrônicos anexados à cópia do *software*, e trazem uma série de termos que regem o uso, a modificação e a redistribuição do *software* pelo usuário-final, além de estabelecer limites de responsabilidade para o distribuidor original. As licenças também costumam instituir o uso condicional do *software*, isto é, se houver violação dos termos de licença, além de ação por danos, cessam os direitos do usuário sobre o *software*.

As licenças de *software* podem ser classificadas em duas categorias principais: as licenças proprietárias e as licenças “*FLOSS*” (Free/ Libre Open Source *Software*).

As licenças proprietárias conferem ao usuário-final alguns direitos de uso sobre cópias de um *software*. Entretanto, não ocorre uma efetiva transferência de propriedade destas cópias para o usuário-final; há apenas uma permissão para determinados usos, de modo que a aceitação da licença torna-se um imperativo para a realização de qualquer finalidade. As licenças proprietárias de *software* podem ser mais ou menos restritivas dependendo das condições de uso que impõem ao usuário-final. Alguns tipos de licença proprietária têm sido classificados como *Quasi Open Source*. Isto significa que, embora haja um componente *Open Source*, ficam também instituídas restrições de uso do *software*, ou ainda obrigações em contrapartida ao uso do código-fonte.

Por sua vez, as licenças “*FLOSS*” efetivamente conferem a propriedade das cópias do *software* ao usuário-final. Isto significa que, no que se refere ao uso da cópia, o usuário-final possui as mesmas prerrogativas do detentor dos direitos autorais, independentemente da aceitação ou não da licença. Mas embora haja transferência da propriedade da cópia, os direitos autorais permanecem com o distribuidor original, de modo que o exercício de alguns direitos exclusivos (como redistribuição, ou distribuição de versões modificadas do *software*) depende da aceitação dos termos da licença.

Assim, tanto as licenças proprietárias de *software* como as licenças do tipo “*FLOSS*” operam segundo o regime de direitos autorais. As licenças “*FLOSS*” correspondem a um aparato legal que institui um regime de colaboração em troca da cessão da propriedade intelectual. A idéia fundamental das licenças “*FLOSS*” é afastar as restrições impostas pelos direitos autorais, garantindo maior liberdade de uso, modificação e redistribuição dos *softwares*; não poderia haver “*FLOSS*” sem propriedade intelectual.

Organizações como a *Free Software Foundation* (“*FSF*”) e a *Open Source Initiative* (“*OSI*”) mantêm definições para “*Free Software*” e “*Open Source Software*”, a partir das quais classificam e certificam os diversos tipos de licença. Tais organizações possuem amplo

reconhecimento na comunidade de programadores, o que lhes garante o status de *certification marks*. Fundamentalmente, as licenças são labeladas como *Closed Source* (ou proprietárias) e “*FLOSS*” (*Open Source* ou *Free Software* dependendo da organização), havendo ainda subtipos dentro de cada uma destas classificações.

Segundo a definição da FSF, para que um *software* seja considerado “*Free Software*”, quatro liberdades fundamentais devem estar garantidas ao usuário: (i) processar o programa para qualquer finalidade; (ii) estudar o funcionamento do programa e modificá-lo para atender a quaisquer outras finalidades; (iii) distribuir cópias do programa a terceiros; e (iv) aprimorar o programa e disponibilizar cópias das versões modificadas ao público. Para que estas liberdades tenham efeito, faz-se necessária a disponibilização do código-fonte do *software*.

Nesta mesma linha, a OSI mantém sua definição para licença “*Open Source*” segundo uma lista de dez requisitos: (i) garantia da liberdade de redistribuição do *software*; (ii) disponibilização do código-fonte do *software*; (iii) liberdade de modificação do *software* e de elaboração de *softwares* derivados, permitindo-se que sejam redistribuídos sob os mesmos termos de licença do *software* original; (iv) a licença poderá restringir a distribuição do código-fonte em uma forma modificada unicamente se permitir a distribuição de *patch files*, com o propósito de modificar o *software* durante sua construção; a licença deve explicitamente permitir a distribuição de *softwares* desenvolvidos a partir do código-fonte modificado; a licença deve determinar que trabalhos derivados carreguem nome e número de versão diferentes do *software* original; (v) vedação de discriminação de quaisquer pessoas ou grupos; (vi) vedação de discriminação do campo de atuação onde se utilizará o *software*; (vii) a licença do *software* deve aplicar-se a todos aqueles aos quais o *software* seja redistribuído, sem a necessidade de implementação de uma nova licença entre as partes; (viii) os termos de licença do *software* independem que este tenha integrado uma distribuição de *softwares* em particular; os termos de uso do *software* sempre seguem a licença original, e aplicam-se a todos aqueles aos quais o *software* tenha sido redistribuído; (ix) a licença de um *software* não se aplica a *softwares* que tenham sido distribuídos em conjunto com este; (x) a comunicação da licença não deve depender do uso de tecnologias, estilos ou interfaces de caráter individual.

Em oposição aos *softwares* controlados por direitos autorais estão os *softwares* em domínio público. Sobre estes não incidem quaisquer direitos de propriedade ou exclusividade; conseqüentemente, não se aplicam instrumentos particulares que regulem sua utilização. Para que um *software* encontre-se em domínio público, é necessário que o prazo legal de vigência

dos direitos autorais já tenha transcorrido, ou ainda que o detentor destes direitos abdique de sua exclusividade, colocando o *software* em domínio público.

Mas o abandono deliberado dos direitos sobre um *software* pode apresentar alguns inconvenientes. Além de representar uma doutrina jurídica ainda pouco clara e de efeitos incertos, não permite ao detentor original dos direitos qualquer controle ulterior sobre a redistribuição do *software*. Nestes casos, o mais comum é que a distribuição opere sob uma licença que aproxime os direitos de uso, modificação e distribuição do *software* aos de um *software* em domínio público. Em outras palavras, através de uma licença pode-se liberar determinadas ações e proibir ou condicionar outras, de modo a se obter o resultado legal desejado. Via de regra, estas licenças são do tipo “*FLOSS*” e podem ser do tipo permissivo, ou do tipo *copyleft*.

As licenças “*FLOSS*” permissivas equiparam as liberdades do usuário a uma situação de domínio público, sem oferecer qualquer limitação ao uso, modificação ou distribuição do *software*. Aqui, a idéia é garantir o máximo de liberdade ao usuário-final, que pode inclusive criar e distribuir trabalhos derivados com licenças proprietárias.

Já as licenças “*FLOSS*” *copyleft* visam proteger o conceito de *software* livre na medida em que impõem uma cláusula “*viral*” ou “*share alike*”. Isto implica que a distribuição de trabalhos derivados deva adotar o mesmo tipo de licença que o *software* original, o que impede que se criem *softwares* proprietários a partir do código-fonte de *softwares* livres.

A *GNU General Public License* (“*GPL*”) é um exemplo de licença do tipo *copyleft* amplamente utilizada. A *GPL* garante ao usuário-final direitos de uso, de realização de engenharia reversa, de acesso ao código-fonte e de adaptação, modificação e redistribuição de cópias do *software* original ou de versões modificadas. Em contrapartida, exige também o cumprimento de determinadas obrigações, como a disponibilização do código-fonte e a garantia dos termos de licença originais quando da distribuição de versões modificadas. O não-cumprimento destes requisitos implica em infringência de direitos autorais. Outros exemplos de licenças *copyleft* são a *Common Public License* (“*CPL*”) e a *European Union Public License* (“*EUPL*”).

Mencionem-se ainda licenças como a *GNU Lesser General Public License* e a *Mozilla Public License*, que também são do tipo “*FLOSS*” *copyleft*, mas que exigem a manutenção da licença apenas para alguns tipos de trabalhos derivados (é o caso de algumas bibliotecas de *software*, que permitem que outros *softwares* vinculem-se a elas sem que tenham que ser distribuídos sob a mesma licença). Este tipo de termo de licença é denominado *linking license*.

Já dentre os exemplos mais comuns de licenças “*FLOSS*” permissivas, podemos mencionar a BDS, a MIT, a X11 e a Apache, que além de garantir as liberdades da GPL, não impõem quaisquer restrições quanto à disponibilização do código-fonte ou à manutenção de uma licença equivalente em trabalhos derivados. Licenças “*FLOSS*” *permissivas* não impedem o desenvolvimento de *softwares* proprietários a partir de *softwares* “*FLOSS*”.

3. Estudo de Caso - LabNUMERAL - Pesquisa, Aplicações e Aprendizagem de Métodos Numéricos em Engenharia – O Uso de Licenças para o Desenvolvimento Colaborativo de *Softwares* em Ambiente Acadêmico

LabNUMERAL (*Laboratory for Numerical Methods in Engineering: Research, Application and Learning*) é um projeto desenvolvido por pesquisadores da Escola Politécnica da Universidade de São Paulo para pesquisa, aplicação e aprendizagem de métodos numéricos em engenharia. Uma das principais vertentes deste projeto é o desenvolvimento de ferramentas computacionais, mais especificamente *softwares* CFD de interesse (aplicativos em *Computer Fluid Dynamics*), incluindo ferramentas de geração de malhas (pré-processadores), ferramentas de visualização de escoamento (pós-processadores) e ferramentas tipo *solvers*. A elaboração, gestão e aprimoramento de tais ferramentas operam por meio da implantação de um ambiente computacional de caráter colaborativo, que oferece suporte ao desenvolvimento das diversas atividades do projeto. Por vias do uso de uma ferramenta do tipo portal, viabilizam-se: a) a intercomunicação entre usuários por fóruns, *chats*, e-mail; b) a construção de conhecimento colaborativo por *wikis*; c) o processamento remoto de ferramentas computacionais técnicas e administrativas; d) a construção de bancos de dados compartilhados com informação incremental baseada em processamento das ferramentas aplicativas; e) a aprendizagem e gestão de conhecimento por meio de ferramentas computacionais; f) a gestão de projeto; g) o controle de acesso; e h) a gestão de segurança.

A principal vantagem de se optar por um *modus operandi* colaborativo é a significativa ampliação da escala de contribuição intelectual ao projeto. Neste mesmo sentido, são utilizadas preferencialmente estruturas de informática do tipo aberto, o que inclui a opção por sistemas operacionais e demais *softwares* com licenças *Open Source*, a aplicação de linguagens científicas abertas e o uso de interfaces e ferramentas de edição do tipo *wiki*.

Com relação à produção de *softwares* CFD de interesse, o laboratório dá prioridade à adoção de licenças do tipo “*FLOSS*” *copyleft*, como a *GNU GPL*, ou licenças *Quasi Open Source* em

dependendo do estágio de desenvolvimento da pesquisa, e dos objetivos específicos de cada vertente do projeto.

Ambos os tipos de licença de *software* podem apresentar vantagens e desvantagens, e a adoção de um ou outro modelo será avaliado no caso a caso. Neste aspecto, Veelo (2005; 2009) é bastante esclarecedor no que diz respeito às vantagens acerca da adoção de licenças “*FLOSS*” no desenvolvimento de *softwares* em áreas de alta especialização tecnológica, notadamente no campo da engenharia naval. Segundo o autor, o uso de *softwares* “*FLOSS*” promove (a) *peer review* (i.é., os usuários atuam como *beta testers*, e deste modo alertam para a existência de *bugs*, corrigem eventuais problemas, incrementam funções existentes e ampliam as possibilidades de ação do *software*. Sob uma licença *copyleft*, a disponibilização destes aprimoramentos deverá reverter para a comunidade segundo o mesmo tipo de licença “*FLOSS*” original); (b) possibilidade do uso de bibliotecas de *softwares* “*FLOSS*”, sem que se tenha de partir do zero para o desempenho de funções já resolvidas em outros blocos de programas (em outras palavras, os objetivos são atingidos mais rapidamente e com menos esforço); (c) predisposição da comunidade científica em contribuir e integrar o projeto, viabilizando o desenvolvimento de sistemas projetuais mais complexos.

Entretanto, em algumas situações, o alto grau de permissividade de licenças “*FLOSS*” pode ter efeitos indesejáveis, sobretudo no que diz respeito à relativa perda do domínio sobre a ferramenta desenvolvida.

Trata-se aqui de ferramentas que resolvem problemas de nichos tecnológicos muito limitados, ou de grupos muito restritos de usuários. Mas embora tais ferramentas não representem grandes expectativas em termos de proventos vindos de *royalties*, ainda pode haver outros tipos de vantagem na aplicação de licenças que limitam o uso do *software*. Como exemplo, mencione-se uma situação em que, durante determinada fase da pesquisa, seja interessante aos desenvolvedores a coleta de dados advindos da aplicação concreta do *software*. Nestes casos, uma licença *Quasi Open Source* poderia trazer prerrogativas como o uso livre e gratuito do *software*, o acesso ao código-fonte e a possibilidade de modificação do código-fonte para usos pessoais, e em contrapartida impedir a redistribuição do produto, além de exigir que os dados resultantes do uso do *software* sejam concedidos para o desenvolvedor original, apenas com a finalidade de auxiliar as pesquisas e incrementar o desenvolvimento futuro do próprio *software*. Em uma fase ulterior, o próprio desenvolvedor do projeto pode considerar mais adequada a aplicação de uma licença do tipo “*FLOSS*”, e liberar o uso do mesmo com grandes benefícios para a comunidade científica.

4. Considerações finais

O trabalho traçou um breve panorama do regime jurídico de propriedade que regula o desenvolvimento de *softwares*, e apresentou algumas das possíveis zonas de interesse para a comunidade científica em aplicar ferramentas de gestão de conhecimento de carácter colaborativo. Neste panorama, foram traçadas algumas das vantagens e desvantagens da aplicação dos diferentes tipos de licença de *software*, mas sem perder de vista o viés colaborativo e plural da construção do conhecimento.

Bibliografia

Free Software Foundation. At: www.fsf.org/ and www.fsf.org/licensing/essays/free-sw.html (access August 2009)

Open Source Initiative. At: www.opensource.org and [www.opensource.org/licenses
www.catb.org/~esr/writings/cathedral-bazaar/](http://www.opensource.org/licenses/www.catb.org/~esr/writings/cathedral-bazaar/) (access August 2009)

Raymond, E. S. *The Cathedral & the Bazaar - Musings on Linux and Open Source by an Accidental Revolutionary*. At: www.catb.org/~esr/writings/cathedral-bazaar/ (access August 2009)

Rosen, L. *Defining Open Standards*. At: www.rosenlaw.com/DefiningOpenStandards.pdf (access August 2009)

Van Lindberg, *Intellectual property and open source - A practical guide to protecting code*, O'Reilly, 2008, 1aEd

Veelo, B. N. *Free Software as an Option for Ship Design*, *Schiffstechnik*, vol. 52, pp. 172–188, 2005.