

THE EVOLUTION OF FREE/LIBRE AND OPEN SOURCE SOFTWARE LICENSES : A DYNAMIC MODEL

Jean-Baptiste Soufron*, Jean Sallantin†

12 mai 2005

Résumé

We address the understanding of the structure of contracts and their evolution via a dynamic model based on the data gathered through the creation of Free/Libre Open Source contracts and their evolution. We then show how this model can be used in order to facilitate licensing negotiation for players.

Introduction

There has been an increasing interest from lawyers and social scientists for the development of legal tools adapted to share knowledge. To this end, several contractual tools have been proposed for allowing the best uses of intellectual property law, notably Free/Libre Open Source Software licenses (or FLOSS licenses). The legal applicability of these licenses have been well studied but there has been no model proposed to explain their creation and their evolution. In this view, it is of utmost interest to propose tools enabling people to understand the processes at work and use them for their best interest in contractual negotiations. Existing approaches in community findings are

either based on expert systems or on classical contractual approach. There has been roughly no attempt to link different licenses between them under a dynamic model. Actually, the various study of FLOSS often concentrates on a very small number of licenses. We give a formal framework for understanding the creation of these contracts and their evolution, using a model based on the realistic approach that contracts are legal answers to problems raised by users, and that new contracts are obtained by combining the clauses of ancient ones or by creating new ones. Suggesting that this leads to the emergence of a legal domain through the abduction of its terms and the induction of the constraints linking them, we eventually propose to precise this model to help people to better negotiate licenses and contracts. Our main source of data is Sourceforge¹, the world's largest Open Source software online repository providing free hosting to nearly 100 000 projects². This data is summarized and publicly accessible on their website but we precise our study using the various mailing-lists associated with the projects. Eventually we add some data from external FLOSS sources when it is available.

*CERSA (Centre de Recherche en Science Administrative), 10 rue Thénard 75005 Paris France, corresponding author: jean-baptiste.soufron@cersa.org

†LIRMM (Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier), 161 rue Ada 34392 Montpellier Cedex 5 France

¹<http://sourceforge.net/>

²following the data available in April 2005, there were 99 803 projects accounting for 1 068 803.

1 Evolutive contracts

1.1 Law in Text

The first examples of Law as a text were quite complicated. Hamurabi's Code for example, contained 282 different rules, each of them being a legal answer to practical problems. It regulated various issues like the right of the poor to seek redress from wrongs committed by the wealthy or by the nobility, the rights of women, including the rights of women to own property in their own names, and even the right to divorce a husband for grounds which would be recognized even today. But only the questions within the list provided by the Code could receive a solution. It was easy to apply but when any new question were to arise, the last article of the Code precised that the king was the only competent authority to answer it. At that time, Law was a complicated set of norms coupled with a top-down evolution process, difficult to change and to individualize.

1.2 Law in Action

Law today offers many other ways to regulate texts just setting out plain rules for citizens and stating that some authority will answer new questions. People are now allowed to create new answers and contracts are a good example as being law in action, created through a normative process emanating from numerous sources interacting to create Agreements.

Contracts Contracts are Law to the parties³. They are agreements containing a com-

³According to Article 1101 of the French Civil Code, a Contract is an agreement by which one or several persons bind themselves, towards one or several others, to transfer, to do or not to do something. Then, as Article 1134-1 puts it, Contracts are only lawfully entered Agreements entered and they take the place of the law

bination of clauses, where each clause can be described as an answer to a given legal problem⁴. As parties can create new clauses or adapt existing ones when they face new problems, contracts can be efficiently described as a source of Law. But when applied, these new clauses will be confronted with legal norms governing the validity of contracts. If they get censored by judges, parties will need to modify them or to create new ones once again. Thus, getting their legitimacy from Law and imposing their content to the judiciary process, contracts also take place into the hierarchy of norms regulating human activity and bringing previousness to human relations⁵.

Evolution In this model, the dynamics of contracts can be described as a continuous evolution where people add and modify clauses to correct errors in the application of a given contract. Needing to be legally formed, Contracts can first be described as a protection for Individuals, controlling these mandatory Agreements with a certain number of limitations and constraints⁶.

for those who have made them. Whether they have a specific denomination or not, they are subject to general rules which are the subject matter of this part of the Code but following Article 1135, they are binding not only as to what they express, but also as to all the consequences which equity, usage or statute give to the obligation according to its nature.

⁴F. Terré, P. Simler, Y. Lequette, *Droit civil Les obligations*, Dalloz, 2002

⁵H. Lécuyer, *Le contrat, acte de prévision*, Mélanges F. Terré, Dalloz 1998, p. 656

⁶Ch. Jamin, *Plaidoyer pour le solidarisme contractuel*, in *Le contrat au début du XXIème siècle, Etudes offertes à J. Ghestin*, L.G.D.J. 2001; D. Mazeaud, *Loyauté, solidarité, fraternité, la nouvelle devise contractuelle?*, in *L'avenir du droit, Mélanges en hommage à F. Terré*, Dalloz, 1999 (dir.de) Ch. Jamin et D. Mazeaud, *la nouvelle crise du contrat*, Dalloz 2003 et (dir. de) L. Grynbaum et M. Nicod.

1.3 Complex contracts

Given that contracts are used on an individual basis, standard economic models imply that they should be highly complicated : rich in the expected number of payoff-relevant contingencies ; variable in the magnitude of payoffs contracted to flow between parties ; and severe in the cognitive load necessary to understand the contract. Yet most real-world contracts are simple on all this points⁷

Complex Systems Instead of being highly complicated, contracts are quite simple texts evolving through dynamics as a complex system controlled by individual constraints imposed by the will of the parties, and legal constraints imposed by the Law⁸. Respecting these constraints is difficult but this evolutive mechanism diminish the cost of legal evolution and leads to more adapted rules for a better justice⁹

FLOSS licenses We will limit our analysis to the example of the evolution of FLOSS licenses and we will try to propose a framework highlighting the evolutionary process at work in contract law. The problem is to understand how players can best negotiate these contracts, and we propose to address four points to understand the dynamics of FLOSS Contracts.

⁷Ch. Jamin, *Économie et droit* in Dictionnaire de la culture juridique, D. Alland and S. Rials (dir.), Paris, PUF, 2003

⁸Ch. Jamin, *à Révision et intangibilité du contrat ou la double philosophie de l'art. 1134 du Code civil*, Droit et patrimoine, mars 1998, p. 58

⁹A. Kronman and R. A. Posner (dir.), *The Economics of Contract Law*, Boston, Little, Brown and Co, 1979 ; E. Mackaay, *Analyse Économique du droit - I. Fondements*, Montréal/Bruxelles, Thémis/Bruylant, 2000 ; E. Mackaay, V. Leblanc, *L'économie de la bonne foi contractuelle*, in Mélanges Jean Pineau, Benoît Moore (dir.), Montréal, Éditions Thémis, 2003, pp. 421-459

First, it is important to describe the mechanism of FLOSS licenses. Then, we will describe the emergence of FLOSS legal domain as a mechanism involving the systematisation of the domain and its application. Finally, we will introduce a tool implementing this model to help writing new licenses.

1.4 FLOSS Principles and Philosophy

Free as Free Speech First of all, Free/Libre and Open Source Software doesn't mean free to use : to briefly precise the principles laid down by the GNU GPL in 1989, a license allows a holder of Intellectual Property Rights to grant the right to use his software to another. The license usually specifies the conditions under which the Software can be used or disseminated to others, payments to the licensor, whether modifications of the software are allowed, the risk and liability each party assumes, representations and warranties, and promises of support and maintenance. Just like any contract, the license must answer a vast number of questions. A source code license grant access to source code for the licensee. Open Source licenses can be defined as a specific set of terms and conditions that meet the requirements summarized by the Open Source Initiative that were only summarized in 1997.

Four requirements FLOSS licenses essentially need to answer four requirements. Each FLOSS license is a different combination of these four rights according to different and specific provisions and the ecology of Rights in an Open Source Software can be represented (Fig. 1). First is the right to access the source code - the right to get the source code and to read it. This right is the main component of any Open Source software since it is a necessary preliminary step to all of them. Every Open Source licenses grant access to the

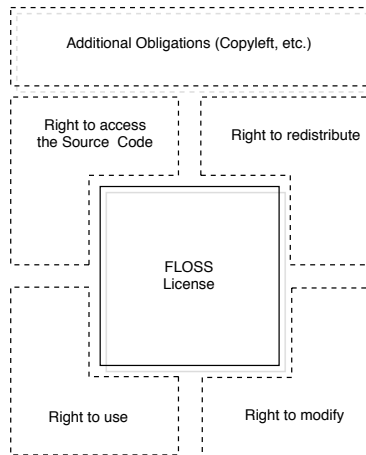


FIG. 1 – The four requirements of FLOSS licenses

Source Code of the Software. The right to modify is the right to change the source code in order to correct or to adapt it. Most Open Source licenses recognizes this right but not all of them. For example the Perl Artistic license states that if a package is modified so that it changes from a Standard Version, then these changes must be posted in a very specific and detailed way described in the license. The right to redistribute, the right to copy and distribute the Software. Licensors often modify this right in many ways. For example, the Sun Community license allows licensees to distribute software only as long as they make no commercial gain directly from the it. Finally, the right to use is the right to compile the source code and execute it. licenses changes this clause the same way they do with the redistribution clause.

1.5 FLOSS Expansion

The primary issues in choosing license terms and conditions are whether the licensor wants revenues, whether it allows modifications to the source code and whether it allows the source code to be forked in different versions.

But most licenses also accept a various number of additional obligations. For example, the Creative Commons NC licenses only authorizes to non-commercial uses of the work.

Users It follows that FLOSS is actually a complex world with a great number of licenses including complex provisions for both parties. One of the first, and the most successful one is the GNU GPL written in 1989, each new license answering new questions following the same legal creation process we are now describing. Since this date, there has been a multiplication of users and since the creation of Sourceforge repository, the number of developers has been growing steadily (Fig. 2).

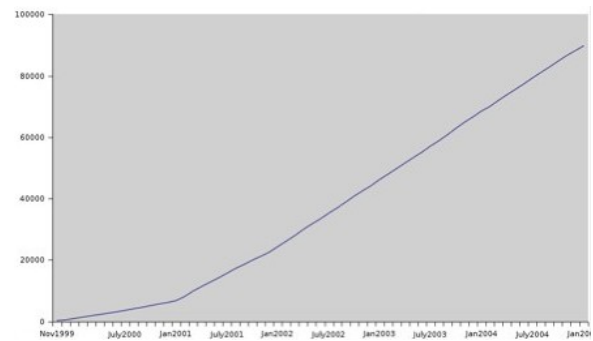


FIG. 2 – FLOSS Developers on Sourceforge between 1999 and 2005 : the number of people using Sourceforge is growing up steadily

licenses But parallel to the multiplication of open source contracts users, there is a multiplication of open source contracts as well, and there are now more than 500 licenses for Software and Content, and new ones are created every day¹⁰. Since 2002 for example, Creative

¹⁰Most of them are listed on the website of the Free Software Foundation, on the website of the Open Source Initiative or on the website of Creative Commons. If we restrict ourselves to FLOSS projects recognized by both the FSF and the OSI, there were 56 FLOSS licenses on

In the meanwhile the number of new projects adding every month is stable(Fig. 3).

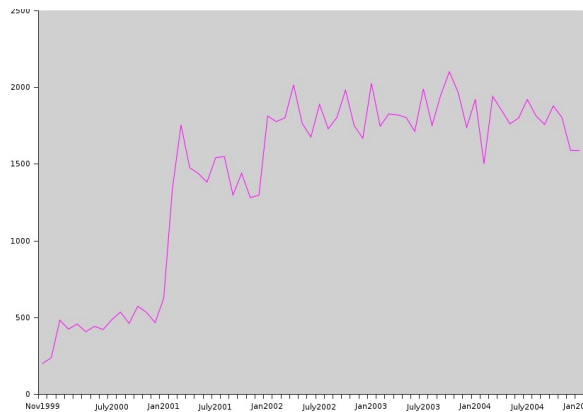


FIG. 3 – New FLOSS projects Sourceforge between 1999 and 2005 : the number of new projects added every month has been stable on Sourceforge since 2002

Commons alone has been producing hundreds of licenses for nearly 30 countries. Today, institutions like the CNRS are also on the process of creating their own range of license under the CECILL label. But users are still demanding more details and more precise modifications. Just as the offer for licenses is growing, the demand of users is also going up. When looking at the licenses used on the Sourceforge repository, we need to read the mailing-list associated with each license to know when they were created (Fig. 4). The demand for new contracts is growing steadily as the number of users is growing. But these contracts are not shared equally amongst projects. For example, the Sourceforge repository is mainly dominated by the GPL and the LGPL(Fig. 5). Even when there is more and more new licenses every year, they only apply to a very small number of projects. New contracts do not answer the questions ruled by existing contracts. They are

Sourceforge in April 2005.

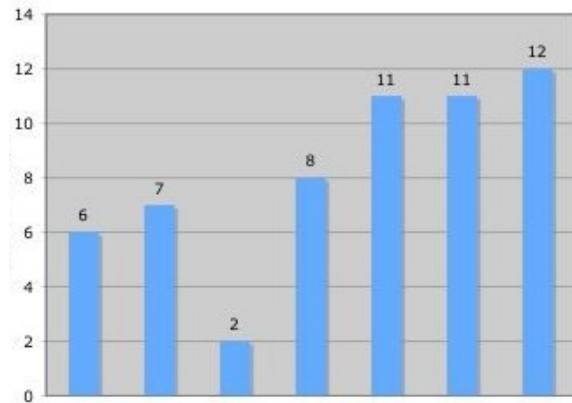


FIG. 4 – New FLOSS licenses between 1998 and 2004 : there are more new licenses created for Sourceforge projects every year

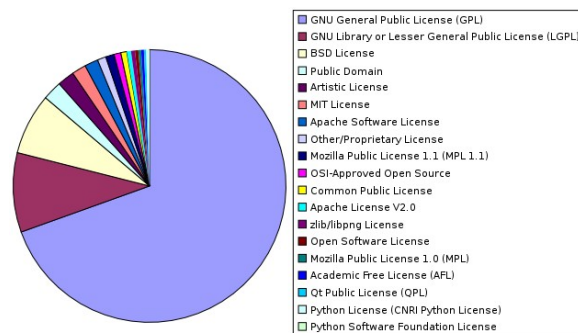


FIG. 5 – Repartition of the more popular FLOSS licenses on Sourceforge in 2005 : the new licenses created every year only account for a very small number of projects

created by modifying old ones in order to rule more new questions. Logically, they concern less users, but the demand for new contracts is also growing up with the number of new users wanting to fit their new needs. Then, we argue that the multiplication of contracts creates a systematization that will be used to better understand the differences between these contracts, the underlying principles of this structure, its legal efficiency, its consistency and its economical interest. We will now address this question of the emergence of the domain of open source contracts.

2 Emergence of the FLOSS legal domain

Simultaneously to the multiplication of contracts, there is a process of systematization consisting in using new licenses to formulate terms to better formulate the legal problems at stake.

2.1 Systematisation

The terms used within the contracts of the domain allow to abduce a vocabulary for this specific domain. Then, it is possible to induce a normative grind stating the causal and logical constraints of the problem while using the same terms defined in the abduced vocabulary.

Application In this model, a FLOSS license are a singular legal solution to a general legal problem and their proliferation is associated to the need for adaptation to real situations. Thus, a legal problem will be described as the virtual abstraction of a real one. Its answer will be given by a set of questions where answers lead to the different clauses as a specific solution. To sum it up, there is an abstraction process consisting in finding the good set of questions which answers will produce the text of the

licenses. In our example of FLOSS contracts, the right to access its Source Code can be seen as a core with every other rights granted by the licensee as optional and potentially modified. Users can also add some other obligations that will add up to the classical rights defining Open Source and Free Software. In the case of the CECILL contract adaptated from the GPL, the right to distribute the software has been divided in two : the right to distribute a modified version of the software and a non-modified one. The CECILL contract will then be a singular solution to a general problem related with the distribution of modified and non-modified versions of a software. This two terms can then become part of the FLOSS legal domain and they will be re-used in following contracts. For example, when people will adapt the CECILL to their own needs.

2.2 Modelizing a class of contracts

Practically, the modelisation of a class of contracts is done by listing every clauses used and by establishing the constraints between them. Each clause will be related to a real use, in the sense that at least one contract is using. We will use a computerized tool called Integre adapted to assist the abduction and induction for a group of lawyers(Fig. 6).

A model of FLOSS licenses As we explained, choosing a FLOSS license means to answer at least the first four different questions we mentionned. But as we can establish constraints between them, each new situation leads to more precise questions, to transformations of the clauses within the existing FLOSS licenses and to the creation of new ones. Asking a specific question means asking another subset of questions, and can exclude from asking another one. For example, accepting to distribute modified versions of a FLOSS Software will first mean to have accepted to let users mo-

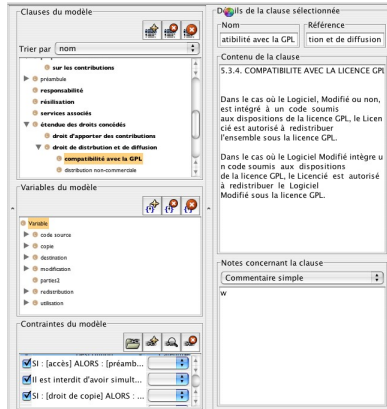


FIG. 6 – Inserting clauses of CECILL-based licenses within integre and detailing constraints between them : the user will hierarchically insert clauses in the first box, he will detail their values in the second one, the constraints in the third one. The text of the clauses are on the left.

dify the Source Code. We can then induce that there is a constraint between these two clauses. Such a work allow to adopt a causal reasoning on the given problem users are confronted to. Each subset of questions will allow us to abduce a doctrinal aspect, and provide the legal referee necessary to contractual freedom. Every question can be seen as a different dimension for solving the problem. Then, summarizing the clauses through an abduction process will allow users to summarize their problem and ask new questions. They will think virtually through this process, and confront its results with reality in their domain : users will refer to the general theory of contracts and to the theory of FLOSS licenses to get answers to their case. Then once again, confronted with reality, these answers will lead them to ask for new questions. When writing new contracts, the goal of users is to be able to attain a better level of legal prediction in and to reduce transfer costs. It is important to describe which ele-

ments of theory are guiding users and how they become more precise with new subsets of questions allowing to create adapted new contracts.

Abduction and induction The abduction process produces an ontology, which means an organization of the term of the domain that can be hierarchical or intricate. Their definition and their coordination through induced constraints will guide their application. It can bring to a better judicial safety and it initiates a co-evolutive process between legal solutions and legal problems. The induction process creates the framework for the resolution of the legal problem. Its principle is based on the research for contradiction followed by its acceptance or its refusal. Allowing a contradiction is equal to create a new contract and make evolve the contractual framework (Fig. 7). Thus, every new contract provides new clauses or modified ones and new questions related to new problems.

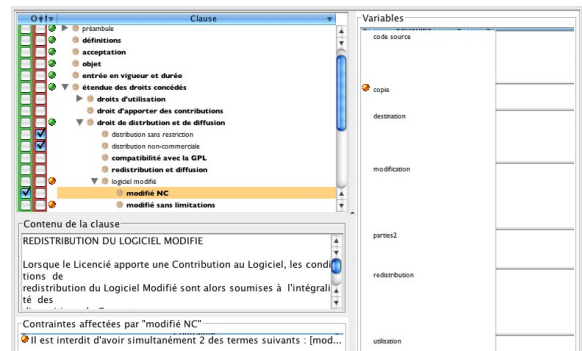


FIG. 7 – Creating a license in Integre by choosing the abducted clauses and following the induced constraints between them : the user decide what clauses to accept or refuse, the values are expressed on the left, the text of the contrat is on the second box and the constraints on the third one

Example In our example, when looking at the mailing-lists of FLOSS projects, most of licenses were created by adding new clauses, or modifying one or several clauses to a previous license. It is a way to go further in the description of the dimensions we previously mentioned. For example, the article 6 of the GPL mandates that any addition to an existing GPL software must itself be released under GPL. But many contributors needed to dynamically link a GPL software library with a classical proprietary software and they felt the GPL was not the correct answer to their problem because of its article 6 since it would mean that they should release their proprietary software under GPL and reveal its source code. Then, the LGPL license was created after the GPL license by modifying its clauses in order to allow people to use external Free Software libraries in their program without being imposed to release the source code of their own software. There were a need to ask new questions to precise the right to use as some people found the solution proposed by the GPL to be not good enough. Thus, users build contracts by putting their clauses together when they answer existing questions, they create new contracts by asking new questions leading to new clauses or modifying the existing ones. For example, people who do not want to allow too many modifications to their software will choose a QPL license which is a modified GPL license stating that additions to the source code must only be submitted as patches and can never be released as binaries. In that case, users felt necessary to ask new questions about the right to modify as the solution brought by the GPL was erroneous in some cases. Also the French license called CECILL is a license drafted by the CNRS and INRIA after the GPL in order to be written in French and more consistent with regards to the French law. A given FLOSS license can be described as a global solution to a given problem, integrating the different sub-solutions to

all of its sub-problems. For example, the Creative Commons licenses are exactly built on this model : they provide a webpage with a given subset of questions where users can determine which contract will best correspond to their needs.

3 Conclusion

Evolutionary model The history of FLOSS licenses tends to show that the number of licenses is always growing. Contrary to the common belief, there is not a need for a unified set of rules, but a need for more clauses describing a greater number of real situations. The more questions are answered by these, the more precisely their combination can describe the reality. In other words, FLOSS licenses answer the judicial problems of software distribution by offering an increasing number of new clauses giving birth to a greater number of licenses. All these licenses are not only law in text but they are also law in action, creating a reasoning field that help to decide how to distribute one's software. It does not become more complicated, but the comprehension of the complexity of reality increase as new questions arise and get new answers.

Producing an ideal The dynamics of FLOSS licenses are not about creation but about adaptation. Adaptation to particular situations under the pressure of objectives and constraints weighing on players. And adaptation to their own history under the need to recreate a consistent general theory of law and to give the right answers for OSS questions. But then, it is interesting to see that these new questions always grow along the four dimensions we mentioned, making them more and more visible. The evolution of the system is following an ideal. The description of this

ideal by the system is actual. Users have real needs, they answer them by interpreting an ideal and asking actual questions. Under classical models, these different solutions would be united under a legal theory giving birth to a set of rules like Hamurabi's 282 rules. But following this model, the development of new contracts will be much more easily done when making it easier for users to ask new questions more rapidly, bringing more than 250 and more contracts. In this case, the equivalent of a legal theory would be the product of the relationship between the actual state of the system, the needs of users put in real situations and the ideal solution. If a contract is the product of a number clauses obtained after answering a number of questions, a legal theory of contracts is the product of a number of contracts obtained after a number of real situations.'

Impact on legal theory From a theoretical point of view, the mechanisms of abduction and induction at work here allow contracts to regulate the social game of players. In fact, the number of different contracts is not exploding with the number of its users, and this bottom-up evolutionary mechanism appears to be of utmost importance for the creation of law. This could be extended : as FLOSS licenses refer and detail a few chosen principles, judges are today referring to human right principles rather than Civil Code articles, leading to new legal solutions on such issues as homosexuality or transsexualism, but also on succession rules or commercial warranties.

Further directions

As a conclusion, this paper considered legal change in FLOSS contracts, focusing on conditions that may determine consolidation or corrosion of their clauses. It would be interesting to study if the stability of one clause is affected

by the number of users choosing it, the flow of recent modifications, the institutional threshold of the need for stability and the weights attached to established provisions and recent new situations¹¹. It should be needed to highlight the role of exogenous shocks, as different dynamic paths may be produced by a similar shock under different contracts. Plus, FLOSS licenses require varying degrees of consistency and this variable, interacting with other exogenous variable, could generate different patterns of evolution. Also, the present model is considered within a unitary judicial system but there are situations in which contracts have an intrajurisdictional effect, rather than an inter-jurisdictional effect across different judicial branches. Finally, this analysis could be scaled further to include the impact of jurisprudence and other forms of regulations in Law.

Acknowledgment

This work has been partially supported by the Normind Company.

¹¹V. Fon, F. Parisi, Judicial Precedents in Civil Law Systems : A Dynamic Analysis, George Mason University School of Law, Law and Economic Working Paper Series 2004