A. Onetti, S. Verma

# Licensing and Business Models

2008/6

UNIVERSITÀ DELL'INSUBRIA
FACOLTÀ DI ECONOMIA

http://eco.uninsubria.it

Printed in Italy in July 2008
Università degli Studi dell'Insubria
Via Monte Generoso, 71, 21100 Varese, Italy

# Licensing and Business Models

Alberto Onetti[*], Sameer Verma[**]

(June 2008)

## Abstract

License affects software companies' business activities. While proprietary software vendors create custom licenses, open source companies have less flexibility. The Open Source Initiative (OSI) defines a list of 72 licenses as open source ("OSI approved"). For a project to follow open source licensing, it has to pick licenses from this set. Logically, we expect that an open source company defines its business model around the license that it selects. Thus, we can assume that business model decisions follow license choice.

In our research we find that in some cases open source companies remove these license constraints for business reasons. We observed cases of open source companies moving from one OSI-approved license to another or companies innovating by adding additional terms. In all these cases, the decision of change is based on the license being a poor fit with their business goals.

Not all open source companies are entitled to change the license because this option is available only to companies that own intellectual property. If they do not, they can try to reshape their business model, but that remains a suboptimal option.

Whether cognizant of it or not, organizations are implicitly choosing a business model when they select a license. Therefore, it is very important to address licensing and business model decisions as one system instead of a disjointed two-step process. For this purpose we introduce (1) an evolutionary model where license selection and business model impact each other and (2) a taxonomy that addresses both licensing and business models. Our approach helps practitioners include revenue considerations in the licensing choice and researchers to more accurately study the antecedents and consequences of license choice.

[*] Associate Professor in Business Administration – Insubria State University – Department of Economics – Via Monte Generoso 71 – 21100 Varese – phone +39 0332 395525 – e-mail aonetti@eco.uninsubria.it

[**] Associate Professor in Information Systems – San Francisco State University – College of Business – 1600 Holloway Ave. – 94132 San Francisco – phone 1-415-338-7016 – e-mail sverma@sfsu.edu

The paper is the result of research carried out by the Authors who jointly elaborated the Abstract. Alberto Onetti wrote the sections 2.4, 2.5, 4, 5.2, 5.4 and 6; Sameer Verma is author of the sections 1, 2.1, 2.2, 2.3, 3, 5.1, 5.3 and 7.

# 1 INTRODUCTION

For open source companies, license is a fundamental element of overall strategy: different licenses have different trade-offs between advantages and disadvantages for different revenue models. Since the license determines what companies can do with their software, license choice strongly impacts the business model and the opportunities of venture funding.

Although license choice represents a critical decision, the choice is not easy. There are a large number of licenses available. The Open Source Initiative (OSI) currently has a set of 72 licenses as open source ("OSI-approved"). The differences between them and their business implications are difficult to grasp.

The issue is further enhanced by the fact that in a majority of cases, the initial license is selected by the creator of the project, who may lack a legal or business background. For example, most people choose GPL or BSD based[1] on their liking for a particular camp (Linux vs. BSD Unix) and stick with it due to popularity (Perens, 2002). Even though contributors claim to be aware of license terms and conditions, as reported in a study by Lerner and Tirole (2002) about GPL, "one might question how understanding and compliance could be garnered from users who do not necessarily purchase the software" (O'Mahony, 2003).

The license governs exactly what companies and developers can do with their software, thus strongly affecting the business model. Therefore, it is not just a development choice, but also a business decision because the licensing terms will define the environment in which a future business may grow and compete. Licensing choice affects the opportunities of funding and acquisition as well: a company's potential investors and any potential buyer will very likely investigate the company's use of open source and find out if the company is in compliance with the license.

Whether cognizant of it or not, organizations are implicitly choosing a business model

[1] Licenses such as GPL and BSD are addressed in the section 2.3.

when they choose a license type. Therefore, there is a need for shifting to a business-driven prescriptive model in license choice.

This need is enhanced by the fact that we are witnessing a strong shift towards a growing commercial participation in the open source arena (DiBona, Stone & Cooper, 1999): “the open source development model has been adopted by new sponsors that include public and private corporations” (O’Mahony, 2007).

On the one hand, we are noticing a slew of commercial companies that emerged from open source projects. Many of them have experienced an infusion of venture capital and some of them have gone through their initial public offering (IPO). Popular examples are RedHat, MySQL, Astaro, Compiere, and Funambol. O’Mahony and Raj (2006) report that “between 2004 and 2005, three times as many venture capital dollars ($300M) went to twice as many firms (33) – the largest spike in venture investments in OSS firms since 2000”.

On the other hand, we see well-established technology companies, such as IBM and Sun Microsystems, launch their previously proprietary software as newly-formed open source projects. Good examples are Eclipse, OpenSolaris, and, more recently, Java. Another variation to the theme is that incumbent proprietary software companies acquire open source start-ups to add open source projects to their portfolio: some appropriate examples are the acquisitions of SleepyCat by Oracle, SuSE by Novell, Xensource by Citrix, MySQL by Sun and Trolltech by Nokia. In parallel, a consolidation trend seems to emerge within the same open source industry with merger and acquisition (M&A) operations involving open source companies: an emblematic case is the acquisition of JBoss by RedHat.

This commercial involvement will likely bring with it a larger share of legal and business understanding. Developers, on the one hand, should be aware of which license protects source code and what happens when code from several sources, protected under several different licenses, are combined to create a product for resale. Company executives, on the other hand, should be aware of what a software license allows and also disallows, even if they do not currently plan to use their software in such a manner.

The goal of this article is to understand the role of licensing of open source and its impact on the different business models that software vendors can adopt. In particular, we will try to understand the business problems the open source companies may encounter with their current license and the opportunity to solve them by changing or “innovating” the license strategy. Prior work has addressed the licensing issue in open source from different perspectives. Among others, Lerner and Tirole (2002) provide an empirical analysis of the determinants of license choice in open source projects; O’Mahony (2003) refers to the licensing terms as a way open source projects protect against the threat of proprietary appropriation; Bonaccorsi and Rossi (2003) examine the license choice of the firms that supply open source products and services and relates it to their structural characteristics and attitudes towards the movement and its community; Välimäki and Oksanen (2005) describe the impact of different open source copyright licensing arrangements on competition in microcomputer operating system markets; Stewart, Ammeter and Maruping (2006) focused on the impact of licensing restrictiveness on the popularity and vitality of open source projects.

This article focuses on the enterprise side of open source software; that is, companies that produce open source software, resell open source software, or provide professional services around open source software as their core activity. Our definition is consistent with Bonaccorsi, Rossi and Giannangeli (2006) who define “as OS those firms that supply, in various ways, OS-based products and services to their customers…even if its offering includes proprietary solutions.” We do not directly address “community-managed open source software projects” (O’Mahony, 2003) nor incumbent software vendors or service providers entering the open source arena, although some of our conclusions may be applicable to these parties. It is worthwhile to single out difficulties in introducing definitions in the open source context given its evolutionary nature. We agree with O’Mahony (2007): “more precise distinctions are needed to communicate the boundaries between firms, nonprofit organization and community”.

The following is a brief description of the sections in this article.

- First, we introduce the topics of open source software in general and its licensing process in Section 2. We also provide a background of the business models in general and present research on some of the implementations of business models in the open source area.
- Next, we discuss the methodology used in this research in Section 3. Our methodology is qualitative in nature and includes primary and secondary data.
- In Section 4 we discuss the relationships between licensing strategy and business model, assuming that they are related in that one influences the other. We present a matrix that combines different categories of licenses with revenue drivers that business models are based on. We identify three main categories of business approaches ("reciprocal", "academic", "dual").
- Next, in Section 5, we use these categories to describe and examine business cases of companies that changed the license or created a new one with additional terms.
- Finally, we discuss the results of business case interpretation and we propose a framework to address the symbiotic process of how licensing strategy and business models influence each other.

We then conclude that this symbiotic observation of "co-evolution" is part of a larger environment which not only includes the corporate perspective but also community-based projects and vendors of proprietary software who co-mingle with open source software. This "ecosystem" (Moore, 1993) also includes the ongoing process of change such as the creation of GPL version 3, which is a significant change from GPL version 2. This environment perspective is extremely relevant but is out of the scope of this article.

# 2 BACKGROUND

## *2.1 Open Source*

Open source software (OSS) is in part developed by freelance programmers who create freely distributed source code by collaborating and communicating over the Internet

(Raymond, 1999; Moody, 2001; Sharma, Sugumaran & Rajagopalan, 2002). Such a collection of open source developers, bug testers, patch submitters, and end-users are often referred to as the open source community collectively. This collection is quite implicit; there is no explicit membership or dues (Perens, 2003). A variety of thinking in the open source world stems from the early days of computing where software was not considered to be the secret sauce – it was simply the code that made hardware perform (Williams, 2002). A majority of the early advocacy and development was done by Richard M. Stallman under the Free Software banner and was later modified slightly as Open Source by a group led by Eric S. Raymond (often quoted as ESR) that wanted better marketing of this software. For the purposes of our discussion we use the term "open source" to represent both free and open source software. About the main differences between these two concepts we refer, among others, to West (2003).

### *2.2 Open Source Licensing vs. Proprietary Licensing*

Proprietary software, including the source code used to create it, is often protected by a patent (Perens, 2005). The patent, not to be confused with copyright, is not free of distribution restrictions and is rarely free of cost. Proprietary software is also defined as the software whose source code is kept secret and belongs to a specific individual or a company (Barahona, Quiros & Bollinger, 1999). In case of proprietary software, the source code is not distributed. These descriptions are largely applicable to proprietary software, although many variations do occur and must be treated on a case-by-case basis. Closely guarded source code becomes the basis of this revenue stream (Verma, 2006).

In the case of open source software, the program includes source code and allows distribution in source code as well as object form. The distribution of source code is governed by an open source license. This license can be selected from a list of pre-approved open source licenses as provided by the Open Source Initiative (Rosen, 2004). A project may have a single open source license or multiple open source licenses. It may also have multiple licenses in the open source and proprietary domains.

### *2.3 License Taxonomy and Distribution*

To better understand the process by which a project adopts a license, it is important to examine the licensing taxonomy of open source software. As opposed to commercial software, open source software creators pick a license from a list of 58 different licenses. These licenses are a collection that has grown from the early days of open source, where many companies such as IBM, Sun Microsystems, Xerox, etc. put forth their own licenses. The Open Source Initiative, a non-profit body, has collected these licenses and certified them as being compliant with the Open Source Definition (Perens, 2003). Such licenses are typically called "OSI approved". OSI has added to its list a number of licenses over the course of eight years or so. Since the launch of the OSI Initiative in November 1998, the number of approved licenses has increased from the initial 25 licenses set to the current list of 58[2].

Despite the large number of licenses, it is important to note that the taxonomy consists of four broad families: "Reciprocal" vs. "Academic" (Rosen, 2004), "Restrictive" vs. "Non-Restrictive" (Stewart, Ammeter & Maruping, 2006), "Viral" vs. "Public" (Hawkins, 2002), and "GPL" vs. "everything else" (Krishnamurthy, 2004). A good example of these distinctions is the General Public License or GPL, under which Linus Torvalds released the Linux kernel in 1991 (Torvalds & Diamond, 2001), versus the classic Berkeley Software Distribution or BSD license, under which the University of California at Berkeley released its code changes to the AT&T UNIX operating system. A third category, sometimes mentioned in taxonomy, is the dual license. However, the dual license is not really a license category – it is a combination of open source and commercial licenses. We will come back to this issue in section 4.3.

To look at how license selection has evolved over the years, we sampled some of the important licenses from 2001 to 2006 at Sourceforge.net (Sourceforge.net, 2006). We

[2] We were able to build a timeline of the license approvals and additions to the OSI list through the analysis of OSI announcements published in the OSI Site History (http://opensource.feratech.com/site_history.php). There were no new additions from January 1999 through the end of July 2001. Afterwards there was a rapid proliferation, with 9 new license approvals in the second half of 2001, 7 in 2002, 11 in 2003, and 7 in 2004. Starting in 2005 the new approvals have reduced significantly.

focused on the most popular and widely used licenses. The decision to narrow the field is consistent with the fact that OSI (2008) declares that, of the set of current 72 licenses in its list, 26 are "considered non-reusable", while many others are "superseded", "redundant", "special purpose licenses", or "voluntarily retired."

| **License** | **2001** | **2006** |
|---|---|---|
| GPL/LGPL | 82.23% | 53.56% |
| BSD | 7.34% | 5.26% |
| Apache | 0.82% | 0.92% |
| MIT | 1.65% | 1.44% |
| MPL | 1.74% | 1.01% |
| Artistic | 3.51% | 1.10% |
| Other | 2.71% | 36.71% |

*Table 1: License distribution 2001-2006*

We have captured the essence of some of these numbers in Table 1. Over a period of time we see interesting trends in the adoption of various licenses.

- GPL remains by far the largest selection. Our data confirm Lerner and Tirole (2002) findings: they stated that of the approximately 40,000 projects they studied in May 2002 at Sourceforge.net, "fully 72% of the licenses are GPL, and its less constraining cousin, Lesser GPL, represents another 10%".
- There is a decrease in the share of popular licenses such as GPL and BSD over a period of time: the market share of GPL/LGPL was at 82% but has now decreased to about 54%; BSD went down from 7% to 5%. They are both a significant loss. We see a similar pattern for other popular licenses such as MPL, MIT, and Artistic.
- The residual category of other licenses, however, has increased from about 3% to almost 37%. Of the 129,000+ projects listed at Sourceforge.net at the end of 2006, about 81,000 have an OSI approved license, while the remaining 48,000 or so have

a non-OSI approved license[3] or have declined to select a license at this time. These data support the evidence of an ongoing process of proliferation of types of licenses[4]; particularly non-OSI approved ones.

We believe that the decline in the use of GPL, BSD and the other most popular licenses and the parallel growth of private or custom (non-OSI approved) licenses is driven by the mismatch between the traditional licenses and the evolving strategic goals of the companies operating in the open source arena. This is due to an increase in the understanding of the nuances of other licenses and a greater participation by the enterprise sector that has taken people away from classic choices to alternative ones. We will focus on this trend in detail in the following sections.

### *2.4 Business Models*

Business model is a concept which is frequently used in the business practice. In the literature the term's adoption began in the late Nineties (Gordijn & Akkermans, 2000; Hamel, 2000; Linder & Cantrell, 2000; Afuah & Tucci, 2001; Galper, 2001; Petrovic, Kittl & Teksten, 2001; Weill & Vitale, 2001; Chesbrough & Rosenbloom, 2002; Gebauer & Ginsburg, 2003; Ostenwalder, Pigneur & Tucci, 2005), but a dominant and widely accepted definition is still missing. Business model is defined in several ways and is used alternatively for describing how a company:

- defines and differentiates its product offerings (value proposition)
- selects its customers and satisfies their needs (market segmentation)
- acquires and keeps customers (market penetration and customer retention)
- goes to the market (promotion and distribution strategy)
- defines the product price (pricing strategy)
- manages its resources (organization)

[3] We refer to non OSI-approved license also as "private" or "custom" license.

[4] OSI has also gone through a process of reduction and de-recommendation of licenses in order to reduce the set of usable licenses that are OSI approved. The institution of the License Proliferation Committee of the OSI ("the LP Committee") is an attempt in this direction. Other efforts, such as the Wizard Project (http://opensource.org/lpc), a web-based wizard to allow people to see which open source licenses meet criteria that they find important, are underway to classify and select appropriate licenses.

- captures profit (profitability).

The business model design is part of the corporate strategy. It is not easy to distinguish the concepts of business model and strategy. For a more complete articulation of the concept of strategy see Bower (1970), Steiner (1979), Tregoe and Zimmerman (1980), and Andrews (1987).

In brief, we can say that defining strategy and designing business models address similar problems in different layers. Formulating a strategy is “developing a broad formula for how a business is going to compete and what its goals should be” (Porter, 1980). A business model “enables an organization to be more effective in creating as well as capturing value” (Chesbrough, 2007). The business model layer would then translate these issues into the policies needed to carry out those goals, i.e. into marketing, sales, and product policies. In Figure 1 we illustrate the different layers using a wheel metaphor: the hub is the strategy (firm’s goals) and the spokes of the wheel represent the business model, i.e. the key policies with which the company aims to achieve these objectives. Strategy and business model are closely interconnected: like a wheel, the spokes (business model) must radiate from the hub (strategy) and reflect the goals, and the spokes have to be connected with each other or the company will not move.

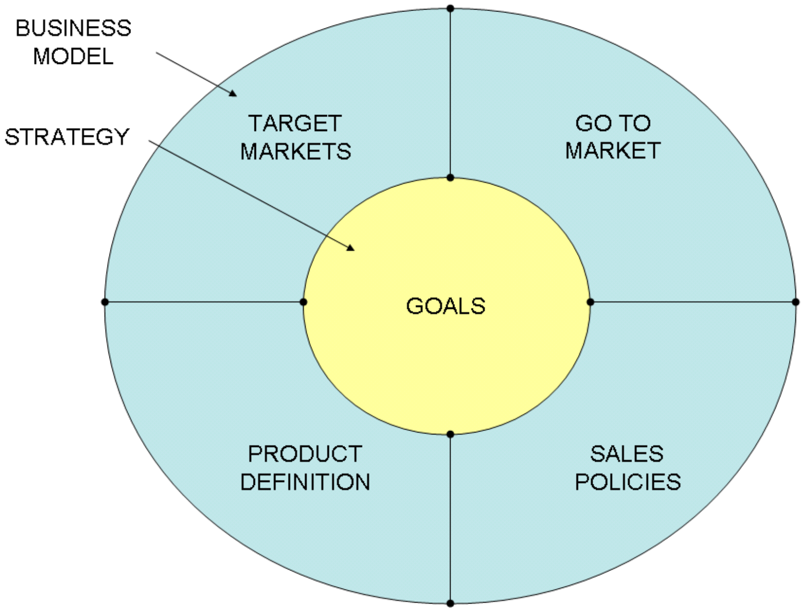

*Figure 1: Strategy vs. Business model*

For the purpose of this article we will refer to a business model as the way a company translates the strategy into business policies. In particular, we will emphasize the mechanisms by which a company intends to generate revenue, focusing on the revenue generation drivers.

### *2.5 Business Models for Open Source Companies*

As open source is increasingly becoming a way to generate business in the software industry, interest about business models for open source has steadily grown. This field has not yet been addressed adequately by the literature. At present, apart from some blueprints from practitioners and consulting firms (see, as an example, the taxonomy proposed by IDC, 2006 or by FaberNovel Consulting, 2007) and some contributions mainly addressing proprietary software-related license models (Hohmann, 2003), there have been few attempts at classification of business models for open source.

The two most prominent and quoted approaches are proposed by Hecker (1999) and by Koenig (2004). Hecker (1999) recognizes eight different approaches for doing business using open source: (1) “Support Sellers”, in which revenue comes from providing service; (2) “Loss Leader”, where a no-charge open product is used for upselling traditional commercial software; (3) “Widget Frosting”, for hardware vendors that use open source software for pushing their sales; (4) “Accessorizing”, in which revenue comes from related products such as books and documentation; (5) “Service Enabler”, with open source software driving usage of revenue-generating on-line services; (6) “Brand Licensing”, where revenue comes from trademarks licensing for derivative products; (7) “Sell It, Free It”, where software products start out their life cycle as traditional commercial products and then are converted to open source; and (8) “Software Franchising”, for companies authorizing others to use its brand names and trademarks in creating organizations doing custom software development in exchange for fees.

Koenig (2004) introduces a classification of seven open source strategies: (1) “Optimization”, i.e. leveraging cost savings achieved on open source to sell added-value, fine-tuned proprietary bricks for specific uses; (2) “Dual licensing”, where a company offers for a fee a commercial version of the open source software with a larger set of features; (3) “Subscription”, where recurrent revenue comes from packaging or bundling open source software along with services; (4) “Consulting”, where a company provides complementary professional services; (5) “Patronage”, using open source to proactively foster the commoditization of a given part of the software stack; (6) “Hosted”, i.e. selling software as a service typically by application service providers; and (7) “Embedded”, where hardware vendors utilize open source as a platform.

More recently, West and Gallagher (2006) examined the activity of firms in open source software to support their innovation strategies. They identify five strategies that firms employ (“pooled R&D”, “product development”, “spinouts”, “selling complements”, and “attracting donated complements”). Although the focus on their research is on the “open innovation” approach (Chesbrough, 2003) and its application to open source, they categorize as “selling components” companies “that sell services for free software”. Companies using dual licensing are included in this category. Additionally, the authors define the dual license model as “a more sophisticated version of selling complements”, based on “a price discrimination (or versioning) strategy”.

It is also worthwhile to refer to two recent papers that explore the business models adopted by open source companies. Chang, Mills and Newhouse (2007), based on a case study analysis, introduces a classification of open source organizations into five types of business models, evolving a categorization proposed by Metcalfe and Rahtz (2007). They identify five models: (1) “Support Contracts”, where a company adopts a support-based model for open source software; (2) “Split Licensing”, for companies adopting a dual license approach; (3) “Community”, that relies on donation and enthusiasm; (4) “Value-added close source”, where a company adopts “pay for software product” and “pay for services” policies; and (5) “Macro R&D Infrastructure”, where the funding comes from research grants. Dallara (2007), using publicly available data, analyzes 80 open source companies and groups them in six main clusters: (1) “Twin licensing”, for companies offering the same software code distributed under GPL and a commercial license; (2) “Split OSS/commercial products”, distinguishing between a basic open source software and a commercial version, based on the community version but with the addition of proprietary plug-ins; (3) “Badgeware”, for companies distributing software under MPL with the addition of a “visibility constraint”, i.e. the compulsory inclusion of visible trademarks, logos, and copyrights notice in the user interface; (4) “Product specialists”, i.e. companies that create or maintain a specific software project and use a “restrictive” open source license to distribute it; (5) “Platform providers”, for companies providing selection, support, integration, and services on a set of projects, collectively forming a tested and verified platform; and (6) “Selection/consulting”, for companies offering consulting and selection/evaluation services on a wide range of projects.

The cited studies represent a significant contribution since they address a gap in the literature about topics such as generating business with open source. Moreover, they provide conceptual frameworks and business case examples.

All the classification approaches described above share some limits:

- on the one hand, they are quite wide-ranging classifications, targeting not only software vendors but also addressing proprietary software and hardware vendors, on-line service providers, and publishing companies. Moreover, they do not focus only on open source companies but also include open source projects as well. Although adopting such a broad approach allows to keep in track all the business activities potentially coming out from the open source arena, the lack of focus results in poor understanding on the actual value drivers a company can effectively leverage;

- on the other hand, there is no clear identification of the criteria on which the classifications are built. The approaches are mainly descriptive in nature, resulting in a list of potential business activities companies and organizations can do around open source software. Additionally, they fail in identifying the core components on which an open source company builds its business strategy.

We believe that adopting a more structured and focused approach could help in understanding the real difference between business models. Moreover it could support empirical analysis and surveys.

## 3 METHODOLOGY

This article is based on a qualitative methodology (business cases) that allows the study of the business in its real context (Eisenhardt, 1989; Yin, 1989) and fits with emerging phenomena (Maxwell, 1996; Padgett, 1998). This is the case of open source and, in particular its recent business orientation: among others, Moczar (2005) talks about "commercial" open source, distinguishing the recent evolution of the movement from its "volunteer" origin. Lerner and Tirole (2005) and O'Mahony (2007) single out the impact on open source from the growing "significant capital investments in open source projects".

We based our research on observing a group of 22 open source companies. The selection process was based on personal references and indirect contacts[5]. The selection of

[5] The authors are active members of various open source software projects and companies. Their involvement

companies was guided by maximizing the variance in perspective of license and commercial approaches. A summary of interviewees is provided in Table 2. We met with senior corporate executives of ten companies and conducted face-to-face open-ended and semi-structured interviews. They spent 1 to 2 hours with us, answering the questions. In two companies, we were able to interview more than one person. As a result, we have involved people operating in the different corporate functions and achieved a broader picture of the company. The interviews covered topics such as the past history and growth of the firm, its success factors, revenue and business model, license choice and policy, anecdotes on the firm's development, relationships with the open source community, and work and management beliefs. Overall, 14 interviews (for about 18 hours) were conducted. The results of our research have also been tested with some of the subjects involved in the interviews in order to increase the external validity of the analytical construct (Yin, 1989).

For the companies we were not able to interview, we collected public information through their web sites and analysis of official documents. For all the companies, we complemented the information collected with public domain discussions from mailing lists, blogs, and opinions from informants and people working in the open source field.

We must point out the limitations of the current study that may impact the generalization of our conclusions. We restricted our sample to a narrow group of open source companies and we addressed them in different ways and with dissimilar levels of analysis. Thus, our ability to generalize may be constrained. These limitations may be addressed in future work by expanding the sample of companies studied. We aim to adopt a broader approach in future research, conducting a structured empirical survey on a larger sample.

in different projects was helpful in obtaining time and attention for interviews.

| | |
|---|---|
| **Geography:** | |
| US Companies | 17 (77%) |
| European Companies | 5 (23%) |
| **License:** | |
| GPL/LGPL | 12 (54%) |
| BSD/Apache license | 5 (23%) |
| Other | 5 (23%) |
| **Business Model:** | |
| Reciprocal | 8 (36%) |
| Academic | 5 (23%) |
| Dual | 9 (41%) |
| **Information collection:** | |
| Face-to-face interviews | 10 (45%) |
| Public information | 12 (55%) |

*Table 2: Interviewees' attributes*

## 4 LICENSING AND BUSINESS MODELS

Open source licenses are a discrete set. For a project to follow open source licensing, it has to pick licenses from the OSI approved set: according to McGowan (2001), "to say a code is Open Source is to say that it is subject to a member of a particular category of licenses". The only flexibility that exists is in the selection and not in the language of the license, although the Open Source definition "does not mandate for a closed set of license" (Bonaccorsi & Rossi, 2003). If "everyone is permitted to draw his own distribution scheme as long as it complies with the criteria of Open Source Definition" (Bonaccorsi & Rossi, 2003), receiving the OSI approval may require some time.

Although business models follow taxonomy (Timmers, 1998), they are a lot more fluid and definable by the company that subscribes to it. Since licenses are a discrete set and business models are more flexible, logically we could expect that a company defines its business model around a license that it selects: the license provides one of the major constraints in a business model for revenue. Many authors agree on this point. Referring to Välimäki and Oksanen (2005) "business model possibilities depend on the licensing choice"; Stewart, Ammeter and Maruping (2006) single out how "these (i.e. license) restrictions may constrain commercialization of OSS applications". Similarly, Chang, Mills and Newhouse (2007) signal how "licensing issues play an influential role to a new project", while Muselli (2007) stresses how "open-source licenses offer limited possibilities in the choice of clauses, since very strict criteria must be respected".

License choice comes first and sometimes it is not a matter of choice at all for many open source companies that are not entitled to change it. For example, EnterpriseDB is a company that produces a relational database management system based on PostgreSQL, an open source database. EnterpriseDB did not create the open source project upon which it is based and therefore it did not choose the BSD license PostgreSQL is distributed under. EnterpriseDB neither controls the copyright nor the license to PostgreSQL, which means licensing is not a matter of choice at corporate level. What EnterpriseDB can do is design a business model consistent with the PostgreSQL license. It did so, creating a superset of PostgreSQL called EnterpriseDB Advanced Server (adding Oracle-compatibility, dynamic performance tuning, and tools), and close-sourcing the code.

Based on these considerations, we can assume that **business model follows license**: open source companies are supposed to shape their business model assuming the license as a constraint.

It is evident that license selection impacts business models. However, the process is not a unidirectional cause-and-effect process. We have noticed that in some instances some open source companies change their license based on the fact that the license is a poor fit with their business activities. We have also noticed that companies go outside the OSI approved open source license set and select or define their own additional terms. We will present and discuss in the next section some of these business cases. In all these cases, we have noticed that these license changes or innovations are driven by business reasons. Business case analysis seems to suggest that sometime **license follows business model**.

Therefore, license selection impacts the business model, but, as the business model requires it, the organization may have to go back and select different licenses or modify the license, thereby giving rise to an evolutionary loop where license selection and business model impact each other.

Our findings support the idea that it is important to consider licensing and business models as one system instead of a disjointed two-step process. Therefore, we introduce a taxonomy that addresses both licensing and business models. Our main goal is not to add a new categorization to the existing ones (see section 2.5), but to draw attention to business implication of licensing choice. We will use this taxonomy as a tool to analyze business cases cited in this article.

Our taxonomy approach is different from what has been done by others for the following two reasons.

- First, we strongly focus on revenue drivers. A viable business model has to generate revenue in a sustainable manner (“no revenue, no party”). Since we focus on the enterprise side of open source software, we believe it is important for open source companies to call attention to the revenue drivers. We consider two main sources of revenue: license fees from sales of software and professional services, including support. Our classification is consistent with Hawkins (2002), who identifies as major potential sources of revenue “sales” and “support”[6].
- Secondly, we provide a simpler and clearer picture of open source approaches to generating sustainable revenue. There are a myriad of licenses available and the business models are typically associated with the different license approaches the companies follow to regulate open source software distribution. The result is a

[6] Hawkins (2002) also categorizes two other sources of revenue: “increased hardware sales” and “value of internal use”. We do not, since our work focuses purely on open source companies that market software products and services and does not address hardware vendors and companies that benefit from the internal use of open source software.

proliferation of business models and related classifications (see section 2.5). However, if we read licenses in terms of business implications, we can tidy up the field by grouping licenses in two categories: “copyleft” and “non-copyleft”. As we discussed in section 2.3, many authors (Hawkins, 2002; Krishnamurthy, 2004; Rosen, 2004; Stewart, Ammeter & Maruping, 2006) assume a similar approach, grouping open source licenses into two families. Obviously, we do not mean that the differences between the licenses are not relevant: each license has its own characteristics that have to be carefully considered. But this is a tactical and not a strategic issue, which can be faced after understanding and selecting the company’s business model. Therefore, we believe that adopting a broad license classification could help in our work of “taxonomization” as well as in empirical analysis. Correspondingly, Bonaccorsi and Rossi (2003) base their empirical survey classifying the licenses used by the respondents in three categories (“copyleft”, “mixed-license”, “non-copyleft”); similarly, Lerner and Tirole (2005) group licenses in three classes (“unrestrictive”, “restrictive”, and “highly restrictive”).

Consequently, the taxonomy is structured along two dimensions.

- The first dimension indexes the company’s **revenue drivers**, i.e. the main source of revenue for the company, ranging from license fees to professional services. Assuming such a perspective, we can distinguish companies adopting a more conventional proprietary software model, where revenue originates mainly from licensing, and from service providers, whose core revenue comes from professional services such as maintenance, support, customization, consulting, and training. Focusing our article on open source companies, we will not take into consideration donations or grants, which represent an ancillary source of income and funding for for-profit organizations.
- The second dimension considers the **license** the company’s business model is based on, distinguishing between copyleft (GPL-like) and non-copyleft (BSD/Apache) licenses.

Jointly considering these dimensions, it is possible to construct a matrix that identifies three main business models that an open source company can adopt (see Figure 2):

- **Reciprocal**: based on copyleft licenses, with revenue coming from professional services
- **Academic**: based on non-copyleft licenses, with revenue originating both from licensing and professional services
- **Dual**: based on copyleft licenses, with revenue coming from license fees

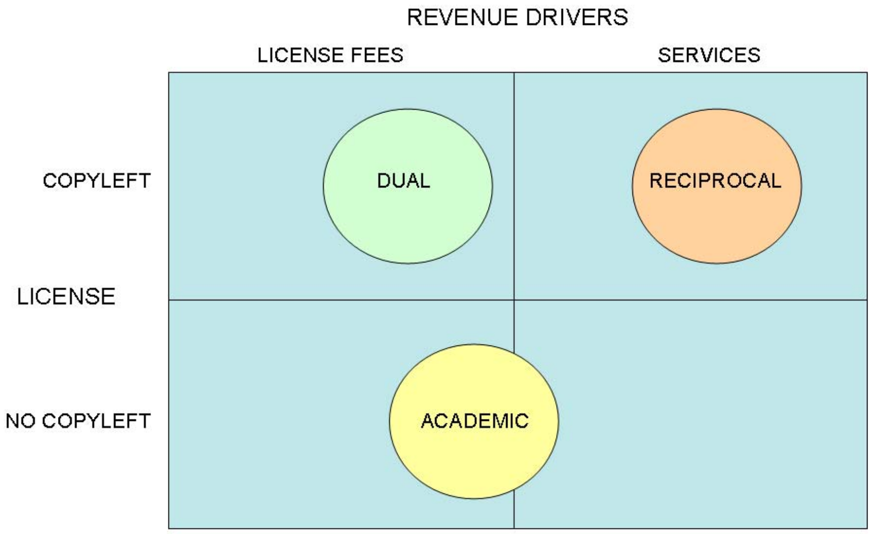

*Figure 2: Open Source Business Model Taxonomy*

### *4.1 Reciprocal*

Reciprocal business models are based on copyleft licenses, i.e. on licenses that allow modification of the code base and the redistribution of the modified version as long as the modified version is distributed under the same license. Copyleft licenses limit commercial adoption of software: some of them, such as the GPL, do it in a very restrictive way ("strong copyleft" licenses), since they require that derivative work also be distributed only under the same license; other licenses, such as the LGPL, MPL, and EPL, are considered "weak copyleft" licenses because they only require modifications to work to be distributed under the same license and may allow re-licensing of the binary form of the work given certain conditions.

What matters to a company adopting a reciprocal business model is that its revenue does not originate from licensing fees. A reciprocal business model is largely a service model, where revenue mainly comes from professional services (maintenance, support, customization, consulting, and training). Some examples of companies adopting a reciprocal business model are: JBoss (acquired by Red Hat), which offers a wide range of support services and tools for its application server; 10X Software, which provides corporate integration consulting for several popular GPL software; and Red Hat, whose revenue comes from Linux maintenance and consulting[7]; Canonical, which provides commercial support (Custom engineering, certification, training and ISV services) for Ubuntu-Linux project.

It is worth pointing out that adopting a reciprocal business model does not mean it is a pure service company. It is possible that some companies offer professional services without contributing to the development of the code. This is typically the case of small and medium system integrators and IT consulting companies. But the most accredited and recognized service providers are those companies that have launched the project and/or developed most of the code. For example, JBoss owned all the intellectual property of its application server licensed under LGPL; Canonical funded the software development and infrastructure services for Ubuntu-Linux project. Therefore, a company adopting a reciprocal business model, in addition to costs for providing customer support and professional services, also has to bear engineering and R&D expenses and investments.

[7] For the Red Hat business model see also McGowan (2001).

### *4.2 Academic*

Academic business models are typically built around a BSD/Apache license. “BSD-style” or “Apache-style” licenses (West & Vitale, 2001) allow companies to use the code in proprietary software, with or without modifications: there is no obligation to propagate the license to derivative work. Code created under these licenses, or derived from such code, may go “closed”, re-branded and be commercialized by anyone (not just the original developers). For companies adopting the academic business model, revenue can originate both from license fees on software reselling and from professional services. In most cases, the BSD license is used only for a core component of the solution (to foster its diffusion), while revenues usually come from proprietary extension, whose intellectual property is owned by the company. Zend, the company behind the highly successful language PHP, is an example of a company with a core open source component (the “Zend engine”) and multiple commercial tools around it. Other examples are Enterprise DB (relational database management system), Greenplum (business intelligence), and Zmanda (backup). All of these companies have a community edition distributed under the BSD license and proprietary commercial extensions.

### *4.3 Dual*

Dual business model is not based on one integrated license. Rather, it is a business policy in which a company that markets a software product gives its customers the choice of two licensing models: open source and commercial/proprietary (Meeker, 2005). A customer can choose to license the software under the terms of an open source license such as the GPL. This allows him to use the software at no charge or sublicense the product's source code to multiple levels of sub-licensees, but it requires re-licensing in source code format, usually on identical terms (i.e. the complete source code for their application has to be available and freely redistributable). This is usually the case of organizations developing open source applications. Alternatively, the customer can choose a commercial license with more conventional licensing terms; in this case, he is required to pay a fee but is freed from the reciprocal provision, being no longer obligated to publish any modification of the source code (Comino & Manenti, 2007). Companies and distributors that do not want to release the source code for their applications as open source software typically ask for the commercial version, because it does not cause their proprietary software to become open source.

Companies adopting the dual business model, not dissimilar to proprietary software companies, base their source of revenue mainly on license fees. MySQL (database servers) is a well-known case, but there are a growing number of companies that are following this licensing approach. Compiere (ERP software application), Db4objects (object database), Funambol (push email and personal information management synchronization), SleepyCat (developers' database), and Trolltech (application development framework and tools) are some good examples[8].

It is not easy to assign open source companies to the business model categories we have introduced earlier because most of them are not "pure play" companies and typically adopt hybrid schemes (Feller, Fitzgerald & van der Hoek, 2002) and (Bonaccorsi, Rossi & Giannangeli, 2006), which are a mix of the above models. We believe that focusing on revenue generation drivers can help in understanding how a company is going to compete and what its business rationale is.

To be considered a company following a dual business model, revenue has to depend heavily on the sale of proprietary licenses. This does not mean that a dual license company cannot sell support and professional services, but the core business has to be built around licensing fees. If services instead of licenses are the mainstream of the revenue, the reciprocal scheme would be a better reference than the dual in designing the company's business model.

Adopting a dual business model is not a choice open to all companies, because it requires intellectual property control. The prerequisite of dual licensing is that the company has

[8] For further analysis about Sleepycat, MySQL and Trolltech business cases we refer to Välimäki (2005). On the Sleepycat experience see also Olson (2006).

control over all the copyrights or, at least, enough control to be able to pick the license terms. This means that the company has to develop all the code in-house or to obtain third-party licenses. External development can be done, but the company needs to ensure that copyrights are assigned from contributors before their contribution is added in. This can meet with significant resistance from the contributor community and drastically drop its contribution to the software development. The main risk for a dual licensing company is to lose the community's support because the company can be perceived as a commercial proprietary software company. Therefore, community effort and public relations programs (in particular a good licensing FAQ) are key (Onetti & Steger, 2007).

Dual licensing requires licensing technically identical product with both proprietary and open source licenses. Frequently, this requirement is not observed by a growing number of open source companies. Many companies offer additional functions and/or services in the commercial version (also called "versioning" strategy), (West & Gallagher, 2006), thus not matching the "same product" criteria. The additional features and services are actually "commercial add-ons", accessible only by paying customers. Some examples of companies selling "ancillaries" are: SugarCRM, that sells a professional version ("Pro") with compelling features not included in the open source version; Funambol, that offers in its Carrier Edition more functions than the Community Edition (i.e., enabling end users to manage device settings and preferences via a web portal that can be integrated into the customer's website); MySQL, whose Enterprise edition comes with the "MySQL Network Monitoring and Advisory Service based on add-on software (not open source licensed) to be installed at the customer's location and not available in the open source community product. Moreover, some companies license the open source software under the terms of custom/private licenses (with their own company name in them) instead of using the GPL or other licenses approved by the Open Source Initiative. Some examples are SugarCRM, Zimbra, and Open Bravo. We will come back to this issue towards the end of our article as we discuss business cases. Right now, it is worth pointing out that, assuming a business model perspective, these companies are strictly comparable to the companies adopting the dual model, although they could be considered as somehow "borderline" to open source principles under a restrictive legal point of view. Therefore, some authors (Stewart, Ammeter & Maruping, 2006) explicitly distinguish what is open source software from what is not by the definition of OSI; i.e., they qualify a software title as open source if its code is released under a license approved by the OSI. However, we believe that our position is supported by the fact that these companies make their products available also as source (that is, in our opinion, what substantially distinguishes open source from proprietary software) and can always change their license, moving from the private license to an OSI approved license. It is, for example, what SugarCRM has done, since it recently announced its decision to adopt version 3 of the GNU General Public License for the next release of its open-source CRM software. Based on the above, this article maintains that companies adding additional terms to OSI approved licenses can be considered as open source players and be included in the dual business model category.

## 5 EVOLUTION OF LICENSING STRATEGY: BUSINESS CASE OF COMPANIES CHANGING OR INNOVATING THEIR LICENSE

There are a growing number of companies changing license during their life, abandoning the previous used license and opting for a new one. We will describe examples of companies moving from BSD to GPL (as with the case of Funambol), and companies following the opposite approach (as done by Vorbis). In both cases, they moved from one OSI approved license to another.

Moreover, other companies are innovating license by adding additional terms to an existing OSI approved license. We will present the cases of SugarCRM, Zimbra, Openbravo, and Funambol, that created their own private licenses.

The cases described below show an interesting trend in changing or innovating license as a way to better accomplish business goals. We will use the taxonomy and the related matrix we introduced earlier in the analysis and discussion of the business cases.

### *5.1 From Reciprocal to Academic*

Many projects start with a reciprocal license in order to gain popularity with the open source community. Reciprocal licenses thrive on the sharing of software and its subsequent propagation after changes have been made. Many open source projects also select a reciprocal license so that the diffusion process gets initiated early.

A good example is the Ogg Vorbis[9] project. Ogg Vorbis is part of the Ogg project, where Vorbis is a way to encode audio files in a manner similar to MP3. However, Ogg Vorbis is not encumbered by royalties. The format is in wide use in games and media players now, but this was not always the case. In its early days, the team that wrote Ogg Vorbis wanted to release its work under a reciprocal license such as the GPL. A reciprocal license would foster sharing and initiate diffusion. However, a reciprocal license does not facilitate adoption by commercial hardware manufacturers, who would much prefer a more “forgiving” license such as BSD. Eventually, in the case of Ogg Vorbis, while most of the code is under the GPL, the project decided to release a variation of their code (called Tremor, a reference decoder that provides an integer-only implementation of the decoder for embedded devices) under a BSD-style license.

In this decision, Ogg Vorbis is able to facilitate diffusion via the GPL and gain market share via the BSD-style license for hardware manufacturers such as iRiver and Samsung[10].

### *5.2 From Academic to Dual*

BSD-style licenses are the less restrictive open source licenses: they allow reselling the code, encouraging commercial development of derivative works, and can coexist with commercial licenses if larger work is distributed under proprietary terms. They represent the best choice for companies that want to reach the largest audience possible.

Academic business model has its weaknesses as well. A company promoting a BSD open

[9] Vorbis is not a company, but its software is widely used in commercial projects.

[10] See http://wiki.xiph.org/index.php/PortablePlayers for a more comprehensive list.

source project typically receives limited contribution from the community. Moreover, a BSD license can create competitors: other companies can launch products based on the code with their own brand name and the project can be forked.

Considering these problems, some companies shifted to more restrictive licenses. This is what Funambol did, moving from Academic to Dual model. When the Sync4j project was launched in 2001, Funambol's target was to have the largest audience possible. Therefore, they chose the BSD license. It worked, attracting top notch developers and lots of users. The project quickly took off. In March 2004, Sync4j could count on more then 2,500 downloads per month. It meant that a large number of people and companies were using Sync4j.

With thousands of people downloading and using the code, you would expect a bit in return: many companies were launching products based on Sync4j, but Funambol did not receive any lines of code in return and little information about the devices tested with Sync4j. Funambol tried everything; including adding a link for contributions on SourceForge, but it did not work.

Then Funambol changed the license to GPL. They could do so because they developed and owned copyrights to all the code. Additionally, they added a commercial license for companies that could not open source their code and give it back to the community.

We note that the change of license has not prevented the Sync4j project (later on renamed Funambol) to grow. Funambol's open source software today counts more than 1,500,000 downloads.

### *5.3 From Dual to Reciprocal*

Dual licensing can sometimes create problems. A lot of potential paying customers, mainly small and medium corporations, use the software without the corresponding license fees. Thus, many companies have switched to a pure service-based model and given up dual

licensing. This choice means refocusing commercial and business activity on support and professional services. The alternative is offering commercial add-ons, as many companies have done. We discussed the "versioning" evolution of dual licensing earlier in section 4.3.
An example of a company moving from a dual to a reciprocal business model is Alfresco that in 2006 went 100% MPL. This choice was not only business driven but was also based on development reasons (they did not like having to constantly figure out what should be "Community" and what should be "Enterprise") and marketing motives (they wanted to have the "we're 100% open source" tagline). This refocusing seems to be successful as they are reporting excellent results in terms of revenue and downloads.

### *5.4 Evolving Dual*

Dual licensing works fine with middleware products, i.e. with products intended to be embedded (Comino & Manenti, 2007). The most successful dual companies, such as MySQL, SleepyCat, and Trolltech, have software products that are meant to be embedded in other applications.

The problem with dual licensing starts with vertical/stand-alone applications. Dual licensing is not suitable for companies because they are not able to impose restrictions on the end user's application code. Stand-alone applications are actually exposed to the so-called "ASP loophole" (i.e., application service providers using the product for distributing services without being forced to open the code.). GPL version 2 ("GPLv2") talks about distribution of software and includes a copyleft clause that triggers when you distribute your code (that is, everything around the code becomes GPL as well). The software industry is experiencing a shift to distributing "software as a service" (SaaS). GPLv2 was created in 1991 before web services were on the horizon and therefore did not address them in the license. As a consequence, some companies interpret distribution of software as a service, not as distribution of software, and use open source software to offer services to the public, without returning anything to the community or paying license fees.

A popular way for closing the ASP loophole, aside from offering compelling features and/or services not included in the open source project and accessible only by paying customers (“commercial add-ons”), is innovating license by adding additional terms to an existing OSI approved license.

A growing number of companies are creating a custom license for handling the problems that dual licensing has dealing with application service providers. SugarCRM, Zimbra, and Openbravo followed a similar approach. SugarCRM introduced the SPL, Zimbra the ZPL, and Openbravo the OBPL, which are all MPL-based private licenses that require corporate branding and permit re-branding only with enterprise license. According to these license terms, all copies of the open source code distributed must include, on each user interface screen, the company’s logo (“Powered by SugarCRM”/“Zimbra Inside”/“Powered by Openbravo”) and the copyright.

Something different has been done by Funambol, which recently modified its licensing terms. While clients continue to be distributed under the GPL, the server is licensed under the Honest Public License (HPL), which is a modified version of the GPL. For closing the “ASP loophole” the HPL expands the concept of distribution to include the use of the software to provide services to the public (SaaS).

This proliferation of private licenses signals the inadequacy of current licensing schemes in sustaining viable business models, particularly the dual approach. There is a need for license schemes able to make open source companies effectively run their business, but the proliferation of private licenses with company names in them is not a sustainable way. In this scenario, the process of rewriting of the GPL license represents an important milestone. On June 29, 2007, the Free Software Foundation published version 3 of the GNU General Public License (“GPLv3”). GPLv3 is a complete rewrite of version 2 of the GPL.

The launch of GPLv3 is a too recent and important event to be addressed properly within a short timeframe. With respect to the “ASP loophole” issue, GPLv3 ultimately did not modify this feature of GPLv2, as conveying does not include the mere execution of the code (CooleyAlert!, 2007). However, GPLv3 does include a clause for compatibility of GPLv3 with the Affero General Public License (“AGPLv3”). AGPLv3 purpose is to impose copyleft on modified versions made available for use over a network[11]. AGPLv3 was recently approved by OSI and this could lead the way to new changes of the licensing polices by open source companies.

Approximately a year after publication, the level of adoption of GPLv3 is continuously evolving. According to Palamida (2008), as of the end of July, approximately 3,000 GPLv2 projects had converted to GPLv3. Less than five months after AGPLv3 approval, approximately one hundred projects moved to it.

Our position, according to the thesis we stated in this article, is that the business circumstances of each company will dictate decisions about the shift to GPLv3 and AGPLv3. If both these licenses prove themselves as viable license schemes for open source business activity, more and more companies will adopt them and move back from private licenses to OSI approved ones. This is the case with Sugar CRM, which recently migrated its license to GPLv3. Funambol has a similar case, where it changed its license from HPL to AGPLv3, right after the OSI approval. If companies are unable to address their business model issues with these newer licenses, more innovation on the license side can be expected.

# 6 DISCUSSION

As previously discussed, it is generally accepted that license selection impacts business models: open source companies typically define their business model around the license. License represents, therefore, a constraint strongly affecting business modeling.

The business cases we presented above show an interesting trend in changing or innovating license that seems to suggest that the reverse pattern (i.e., business model affecting license choice) sometimes applies.

We found companies moving from one OSI approved license to another. As described,

[11] Section 13 of GPLv3 gives express permission to link or combine any GPLv3-covered work with a work licensed under AGPLv3 into a single combined work and to convey the resulting work. GPLv3 will continue applying to the relevant part of the combined work, while AGPLv3's special requirements will apply to the combination as such.

Funambol went from BSD to GPL. Similarly, Vorbis decided to go from GPL to a BSD-style license. In both cases, since the licensing scheme did not properly support the business objectives, the company changed the license, picking one from the OSI license list.

In the cases of SugarCRM which created the SPL (Sugar Public License, i.e. MPL + additional terms), Zimbra with the ZPL (Zimbra Public License), Openbravo with the OBPL (Openbravo Public License), and Funambol with the HPL (Honest Public License), we found that the companies "innovated" the license, adding terms to an OSI-approved license scheme. The decision to create a private/custom license was based on the fact that the licenses included in the OSI approved list are a poor fit with the business activities of the investigated companies.

More recently we have witnessed cases of companies moving from private/custom licenses to OSI-approved ones. Some of them (SugarCRM and Funambol) are companies we mentioned earlier as "license scheme innovators" by building their own license scheme. This may be indicative of the newer OSI licenses addressing current business needs, but also of a stronger underlying dynamic of licensing policy change across commercial open source companies.

From business case analysis we conclude that the changes in licensing strategy we found are mainly "business driven". In all the cases we discussed, the decision to change/innovate the license followed a previous evaluation of the business implications of the current licensing scheme. Since the present license scheme was not able to support their business model (i.e., it was not consistent with the variables defining their business approach as shown in Figure 1), the companies changed or modified it.

Obviously, not all open source companies are entitled to change the license. This is allowed only to companies owning or being able to manage (through copyrights assignation from contributors) the intellectual property rights of the project. If they do not, they can try to reshape their business model, but that remains a suboptimal option.

Business case analysis provides evidence of a **feedback loop** from the business model side going back to the selection criteria of one or more licenses. License selection impacts the business model but, as the business requires it, the company, if it is allowed to, can change or modify the license. Therefore, the process is not a unidirectional cause-and-effect process but a continuous process of adjustment. We have represented this "symbiotic loop" effect in Figure 3: license choice influences business model definition, but, as the business conditions evolve, the organization may have to go back and modify the license or select different ones, thereby giving rise to an evolutionary loop where license selection and business model impact each other.

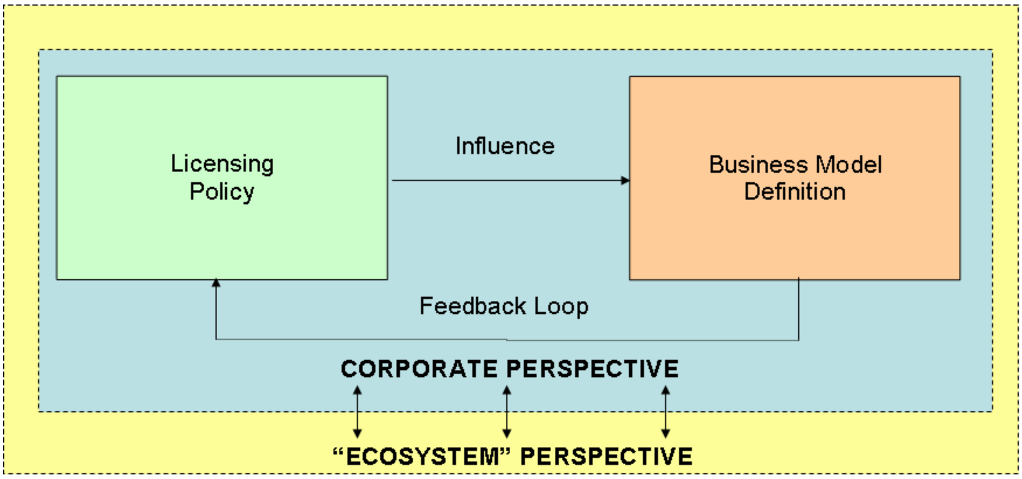

*Figure 3: The Business Model-Licensing Loop*

This symbiotic observation of "co-evolution" is part of a larger environment which not only includes the corporate perspective but also community-based projects and vendors of proprietary software who co-mingle with open source software. Open source companies are embedded in a complex "ecosystem" (Moore, 1993). Thus, to understand a firm's behavior, it is essential to consider this level of analysis (Poole & Van de Ven, 2004).

Adopting such a perspective, we observe broader dynamics both on the business and the license side.

- On the one hand, we can refer to the growing commercial participation in the open source arena we discussed in the introduction. Large established incumbents increasingly embracing open source software as well as M&A involving open source companies are all phenomena strongly modifying the business environment where open source companies operate.

- On the other hand, events, such as the creation of GPLv3 and the approval of AGLPv3, can change the picture on the licensing side, providing open source companies with more options and triggering license change by open source companies.

Obviously, this evolution at system level strongly impacts decisions at corporate level, therefore affecting the loop itself. We recognize the importance of these issues as being influential on licensing strategy and business model definition but have not included them in the scope of our research[12].

The above mentioned considerations suggest the importance of addressing license and business model issues as a system instead of as a disjointed two-step process. License choice should be done considering business implications and business model definition should be done evaluating license constraints.

The business cases we studied prove that for some open source companies the initial license choice was not the best one in terms of leveraging their business activity. These companies had the opportunity to make modifications on the license side because they controlled the intellectual properties rights. However, we believe that there are plenty of open source companies for whom this option is not available. Therefore, they are forced to adopt sub-optimal business models, and this can thwart a company's commercial effectiveness.

## 7 CONCLUSION

This article aims at shedding light on licensing issues for open source companies. Given the growing business orientation of the open source movement (DiBona, Stone & Cooper, 1999; O'Mahony, 2007), we have chosen to assume a business perspective; i.e., we have discussed the impact of licensing on different business models adopted by software vendors

[12] Intellectual property and patents have grown to be a significant issue with respect to the free and open source software industry. In the current paper, however, we do not address these issues outside of their impact on licensing.

that operate within the open source software space.

More specifically, we have classified open source business models, analyzing open source licenses in terms of revenue drivers and business implications. In doing so, we have singled out the implications of license selection for an open source company: since terms of a license determine what companies can do with their software, organizations are implicitly choosing a business model when they select a license type.

We find that open source companies evolve their licensing strategy. A growing number of companies change their license(s) during their business life cycle, abandoning the previously used license and opting for new ones.

Our analysis suggests that the licensing evolution we are facing is business driven. Since open source is becoming a way to generate business in the software industry, software companies are increasingly examining whether their current licensing scheme fits their business models. If a license can prevent growth or create obstacles for business activity, open source companies may change or modify it. If a company finds a suitable licensing scheme from the OSI approved list, it may adopt it. If the company does not find a suitable license, it may "innovate" the license, creating a custom/private license by adding terms to an OSI approved license scheme.

An aspect of strong significance is the party that can actually change the license. This happens to be the party that holds the intellectual property rights to the project. We mentioned that companies that face problems with their current license scheme move to a different one. This change is not always possible, because this option is available only to companies that own intellectual property or gather consensus from all contributors (not an easy option in practice). For example, all projects created under the GNU umbrella require that copyright be transferred to the Free Software Foundation (O'Mahony, 2003). This allows the Free Software Foundation to make licensing decisions. For other companies, the license represents a constraint that can be hard to remove. This makes the choice of the license a critical decision and may spell success or failure for the company and affect the adoption of the product.

Thus, understanding the implications of license selection early in the development process is really crucial because those terms define the parameters in

which the company will compete and succeed. This is not an easy task, if we consider the fact that the choice of the license is typically made by the creator of a project (usually a software developer) who rarely has the skills to understand the business and legal implications of software licenses he adopts. However, it is an issue that has to be addressed as open source business continues to evolve.

Adopting the “feedback loop” approach we introduced could help open source companies’ managers decide if their current licensing policy is consistent with their business model. It is also helpful to software developers as they gear up to launch a new open source project.

Our research is limited in its scope. We address only open source enterprises and service providers that support open source software. However, the issue of licensing strategy and its influence on the business model definition has implications beyond the corporate level. Open source companies are part of a larger “ecosystem” that includes non-profit consortia and proprietary software and hardware vendors who choose to work with open source software on the grounds of interoperability (Verma & Jin, 2004). Within this ecosystem, we also observe significant dynamics on the licensing side, such as the creation of GPLv3 and the approval of AGPLv3. Among other things, GPLv3 addresses the issues of the “ASP loophole”, “Tivoisation”, and patent protection as with the Microsoft-Novell agreement. We recognize the importance of these issues as being influential on the business model definition and licensing selection but have not included them in the scope of our research.

## References

AFUAH, A. & TUCCI, C.L. (2001). *Internet business models and strategies*. Boston, MA: McGraw-Hill.

ANDREWS, K.R. (1987). *The concept of corporate strategy*. Homewood, IL: Irwin.

BARAHONA, J.M.G., QUIROS, P.H. & BOLLINGER, T. (1999). A Brief History of Free Software and Open Source. *IEEE Software*, 16(1), 32-34.

BONACCORSI, A. & ROSSI, C. (2003). Licensing schemes in the production and distribution of open source software. An empirical investigation. *Working Paper*, Sant'Anna School of Advanced Studies, Pisa, Italy.

BONACCORSI, A., ROSSI, C. & GIANNANGELI, S. (2006). Adaptive entry strategies under dominant standards - hybrid business models in the open source software industry. *Management Science*, 52(7), 1086-1098.

BOWER, J.L. (1970). *Managing the resource allocation process*. Boston, MA: Harvard Business School Press.

CHANG, V., MILLS, H. & NEWHOUSE, S. (2007). *From open source to long-term sustainability: review of business models and case studies*. Proceedings of All Hands Meeting 2007. Chang, V. (ed.). Nottigham, UK, OMII-UK Workshop.

CHESBROUGH, H. & ROSENBLOOM, R.S. (2002). The role of the business model in capturing value from innovation: evidence from Xerox Corporation's technology spin-off companies. *Industrial and Corporate Change*, 11, 529-555.

CHESBROUGH, H.W. (2003). *Open Innovation*. Boston, MA: Harvard University Press.

CHESBROUGH, H.W. (2007). Why Companies Should Have Open Business Models. *MIT Sloan Management Review*, Winter, 22-28.

COMINO, S. & MANENTI, F.M. (2007). Dual Licensing in Open Source Software Markets. *MIT Working Papers*.

COOLEYALERT! (2007). *Free Software Foundation Publishes GPL Version 3*. Cooley Godward Kronish LLP.

DALLARA, C. (2007). Business models in FLOSS-based companies. *MIT Working Papers*.

DIBONA, C., STONE, M. & COOPER, D. (ed.), (1999). *Open Sources: Voices for the Open Source Revolution*. Sebastapol, CA: O'Reilly.

EISENHARDT, K.M. (1989). Building theories from case study research. *The Academy of Management Review*, 14(4), 532-550.

FELLER, J., FITZGERALD, B. & VAN DER HOEK, A. (2002). Open source software engineering. *IEE Proceedings - Software*, 149(1), 1-2.

GALPER, J. (2001). Three Business Models for the Stock Exchange Industry. *Journal of Investing*, 10(1), Spring, 70-78.

GEBAUER, J. & GINSBURG, M. (2003). The US Wine Industry and the Internet: An Analysis of Success Factors for Online Business Models. *Electronic Markets*, 13(1).

GORDIJN, J. & AKKERMANS, J.M. (2000). *Business Modeling is not Process Modeling*. International Workshop on Conceptual Modeling Approaches for e-Business, Salt Lake City, Utah.

HAMEL, G. (2000). *Leading the Revolution*. Boston, MA: Harvard Business School Press.

HAWKINS, R.E. (2002). The Economics of Free Software for a Competitive Firm. *MIT Working Papers*.

HECKER, F. (1999). Setting Up Shop: The Business of Open-Source Software. *IEEE Software*, 16(1), 45-51.

HOHMANN, L. (2003). *Beyond software architecture: creating and sustaining winning solutions*. Boston, MA: Addison Wesley Professional.

IDC (2006). *IDC's software taxonomy 2006 and Japan software 2006-2010 preliminary forecast*.

KOENIG, J. (2004). *Seven open source business strategies for competitive advantage*. Available at: http://www.itmanagersjournal.com/articles/314?tid=85 Last accessed January, 2008.

KRISHNAMURTHY, S. (2004). *An analysis of open source business models*. In Feller, J.B. Fitzgerald, Hissam, S. & Lakhani, K. (ed.). *Making sense of the Bazaar: perspectives on open source and free software,* MIT Press.

LERNER, J. & TIROLE, J. (2002). Some Simple Economics of Open Source. *Journal of Industrial Economics*, 52, 197-234.

LERNER, J. & TIROLE, J. (2005). The Scope of Open Source Licensing. *Journal of Law, Economics, and Organization*, 21(1), 20-56.

LINDER, J.C. & CANTRELL, S. (2000). *Changing Business Models: Surveying the Landscape*. Institute for Strategic Change, Accenture.

MAXWELL, J.A. (1996). *Qualitative research design and interactive approach*. Thousand Oaks, CA: Sage Publication.

MCGOWAN, D. (2001). Legal implications of Open Source software. *University of Illinois Law Review*, 1, 241-304.

MEEKER, H. (2005). *Dual-Licensing Open Source Business Models. Mixing proprietary terms with the GPL*. SYS-CON Media.

METCALFE, R. & RAHTZ, S. (2007). *What is open source software?*. OSS Watch/JISC Communications Team. Available at: www.jisc.ac.uk/publications Last accessed July 2007.

MOCZAR, L. (2005). *The open source monopoly*. Available at: http://www.galatea.com Last accessed January 2008.

MOODY, G. (2001). *Rebel code: Linux and the open source revolution*. London, U.K.: Penguin.

MOORE, J.F. (1993). Predators and prey: a new ecology of competition. *Harvard Business Review*, 71(3), 75-86.

MUSELLI, L. (2007). Business models et rémunération des éditeurs de logiciels open source - La mutualisation: un business model original, presented at the Conférence “The diffusion of FLOSS and the Organisation of the Software Industry: From Social Networks to Economic and Legal Models”, Nice-Sophia Antipolis, May, 31 and June 1, 2007.

OLSON, M. (2006). *Dual Licensing*. In DiBona C., Canese, C. & Stone, M. (ed.). *Open Sources 2.0: The Continuing Evolution*. O'Reilly Media.

O’MAHONY, S. (2003). Guarding the commons: how community managed software projects protect their work. *Research Policy*, 2003, 32(7), 1179-1198.

O'MAHONY, S. (2007). The governance of open source initiative. *Journal of Management and Governance*, 11(2), 139-150.

O'MAHONY, S. & RAJ, N. (2006). *Mozilla: the rebirth of Firefox*. Harvard Business School Publishing, Boston, MA: Harvard Business School Cases 907015.

ONETTI, A. & STEGER, H. (2007). The Melting Pot. Segmenting today's open source community. *Open Source Journal (EOSJ)*, August/September, 14-20.

OSI LP (2007). *Committee Report of License Proliferation Committee*. Open Source Initiative. Available at: http://opensource.org/proliferation-report Last accessed July 2008.

OSTERWALDER, A., PIGNEUR, Y. & TUCCI, C.L. (2005). Clarifying Business Models: Origins, Present, and Future of the Concept. *Communications of AIS*, 15.

PADGETT, D. (1998). *Qualitative methods in social work research: Challenges and Rewards*. Thousand Oaks, CA: Sage Publication.

PALAMIDA (2008). *GPLv3 and LGPLv3 information site*. Available at: http://gpl3.palamida.com:8080/ Last accessed April 2008.

PERENS, B. (2002). *The future of open source software*. Boundaryless Information Flow: The Role of Open Source, Cannes, France.

PERENS, B. (2003). *The Open Source Definition*. Available at: http://perens.com/OSD.html Last accessed January 2008.

PERENS, B. (2005). *The problem of software patents in standards*. Cyber Security Policy

Research Institute, George Washington University.

PETROVIC, O., KITTL, C. & TEKSTEN, R.D. (2001). *Developing Business Models for eBusiness*. International Conference on Electronic Commerce, Vienna, Austria.

POOLE, M.S. & VAN DE VEN, A.H. (2004). *Theories of organizational change and innovation processes*. Handbook of Organizational Change and Innovation. New York: Oxford University Press, 374-397.

PORTER, M.E. (1980). *Competitive strategy: techniques for analyzing industries and competitors*. New York: The Free Press.

RAYMOND, E.S. (1999). *The Cathedral and the Bazaar*. Sebastopol, CA: O'Reilly & Associates.

ROSEN, L. (2004). *Open source licensing: software freedom and intellectual property law*. Upper Saddle River, NJ: Prentice Hall.

SHARMA, S., SUGUMARAN, V. & RAJAGOPALAN, B. (2002). A Framework for Creating Hybrid-OSS Communities. *Information Systems Journal*, 12(1), 7-26.

SOURCEFORGE.NET (2006). *Software Map*. Available at: http://sourceforge.net/softwaremap/index.php Last accessed January 2008.

STEINER, G. (1979). *Strategic planning*. New York: The Free Press.

STEWART, K.J., AMMETER, A.P. & MARUPING, L.M. (2006). Impacts of license choice and organizational sponsorship on user interest and development activity in open source software projects. *Information Systems Research*, 17(2), 126-144.

TIMMERS, P. (1998). Business models for electronic commerce. *Electronic Markets*, 8(2), 3-8.

TORVALDS, L. & DIAMOND, D. (2001). *Just for fun*. London, UK: Texere.

TREGOE, B. & ZIMMERMAN J. (1980). *Top management strategy*. New York: Simon & Schuster.

VÄLIMÄKI, M. & OKSANEN, V. (2005). The impact of free and open source licensing on operating system software markets. *Telematics and Informatics*, 22, 97-110.

VERMA, S, (2006). *Software Quality and the Open Source Process*. In Duggan, E. & Reichgelt, J. (ed.). *Measuring Information Systems Delivery Quality*. Hershey, PA: Idea Group, 291-310.

VERMA, S. & JIN, L. (2004). *Diffusion and adoption of open source software within the open source community*. Decision Sciences Institute, Boston, MA.

WEILL, P. & VITALE, M.R. (2001). *Place to space: migrating to ebusiness models*. Boston, MA: Harvard Business School Press.

WEST, J.W. (2003). How open is open enough? Melding proprietary and open source platform strategies. *Research Policy*, 32, 1259-1285.

WEST, J.W. & GALLAGHER, S. (2006). Challenges of open innovation: the paradox of firm investment in open-source software. *R&D Management*, 36(3), 319-331.

WILLIAMS, S. (2002). *Free as in freedom: Richard Stallman's crusade for free software*. Sebastapol, CA: O’Reilly.

YIN, R.K. (1989). *Case study research: design and methods*. Newbury Park, CA: Sage Publication.