ARTICLE IN PRESS

Information Economics and Policy xxx (2011) xxx–xxx

Contents lists available at ScienceDirect

Information Economics and Policy

journal homepage: www.elsevier.com/locate/iep

Dual licensing in open source software markets ☆

Stefano Comino [a,*], Fabio M. Manenti [b]

[a] Dipartimento di Scienze Economiche e Statistiche, Università di Udine, Via Tomadini 30/A, 33100 Udine, Italy
[b] Dipartimento di Scienze Economiche "M. Fanno", Università di Padova, Via del Santo 33, 35123 Padova, Italy

ARTICLE INFO

Article history:
Received 26 April 2010
Received in revised form 16 June 2011
Accepted 5 July 2011
Available online xxxx

JEL classification:
L11
L17
L86
D45

Keywords:
Open source software
Open source business models
Embedded software
Dual licensing
Versioning

ABSTRACT

In this paper we present a theoretical model to study the characteristics and the commercial sustainability of dual licensing, an open source (OS) business strategy that has gained popularity among software vendors. With dual licensing, a firm releases the same software product under both a traditional proprietary license and an open source one. We show that the decision to employ a dual licensing strategy occurs whenever the feedbacks of the open source community are valuable enough compared to the quality of the software that the firm is able to develop in-house. Our analysis points to the central role of an appropriate managing of OS licenses in order to balance the pros and cons of "going open source" and to make this versioning strategy viable for software vendors; our analysis also suggests a possible explanation for the observed proliferation of open source licenses.

1. Introduction

Until recently, open source (OS) has been seen unfamiliar by the business community and, in many cases, it has been perceived as a real threat by commercial vendors. In the very last years, things have changed substantially and both large established incumbents such as IBM, HP or NEC as well as start-ups are increasingly embracing OS strategies.

Commercial firms may enjoy several benefits by "going open source". A firm may take advantage of the contributions of the community of OS developers either in the direct form of code enhancements or in terms of educated feedbacks and reviews received from expert users.[1] Furthermore, open source represents a powerful channel of software distribution: it may constitute a key strategic instrument to improve the perceived quality of the product

☆ Much of this work has been conducted while the authors were visiting the School of Information of the University of California at Berkeley. We are extremely grateful to Hal Varian for his hospitality. Financial support from "Progetto di Ateneo" – Padova, 2005–2007. Paper presented at the 2nd FLOSS international workshop on Free/Libre open source software – Rennes 2008, at the 3rd annual conference of the EPIP association – Bern, 2008, and at the 36th annual conference of the EARIE – Ljubljana, 2009. The authors are grateful to Mark Schankerman, Laura Vici, Marcus Wagner and Federico Boffa for helpful suggestions and comments on earlier versions of the paper. A special thank to two anonymous referees and to the Editor of *Information Economics and Policy* for their insightful comments.

* Corresponding author. Tel.: +39 0432 249211; fax: +39 0432 249229.
E-mail addresses: stefano.comino@uniud.it (S. Comino), fabio.manenti@unipd.it (F.M. Manenti).

0167-6245/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.infoecopol.2011.07.001

[1] It deserves to be noticed that "non-code contributions" from the OS community are as important as "code contributions". For instance, Jullien (2006), in a study on the Open Cascade project, reports that: bug-fixing, preparation of documentation or tutorials and other contributions not directly linked with code writing represented the 20% of the value of the software. Similar findings, for other OS projects, are in Seigo (2006) and in Mueller (2007).

and to enlarge the installed base of users, thus helping firms in establishing an industry standard.

In a recent study based on 218 companies that were collecting at least 25% of their revenues, directly or indirectly, from open source, Daffara (2009) observes that the most common OS business strategies fall into two main categories. The sale of services that are complementary to the open source software, such as customization, consulting, training and documentation, constitutes the most common category of OS business strategies. The second most popular category is versioning. Firms employing this business strategy offer two versions of the software, an OS version and a proprietary (non-OS) one; the proprietary version usually includes upgraded packages and additional functionalities that increase the perceived quality of the software. Within this second category, a business strategy which is peculiar to the software industry and that it is becoming increasingly popular among commercial vendors is *dual licensing*; according to Daffara (2009), around 10% of the companies considered in the sample dual license their code. Notable examples of software packages released according to this commercial strategy are MySQL, Berkeley DB, Qt, and Asterisk (for details see Välimäki, 2005; Moody, 2006).

With dual licensing firms mix traditional and OS-based strategies by offering the same software product under both a traditional proprietary license and an open source one; in the latter case, the software is typically provided for free or at a nominal fee. There are various reasons why customers, when offered a free OS version of a software, may still prefer to pay for the proprietary version; certainly, one of the most important reason accrues to the reciprocal provision imposed by some OS licenses: open source customers are required to redistribute their derived works under the same licensing scheme as the original software, including the requirement to make the source code of the derived software publicly available.[2] To better grasp this critical issue, it is useful to quote Oracle, the vendor of the embedded database BerkeleyDB; in its web page, Oracle describes its dual licensing strategy as follows:

> "Our open source license permits you to use Berkeley DB [...] at no charge under the condition that if you use the software in an application you redistribute, the complete source code for your application must be available and freely redistributable under reasonable conditions. If you do not want to release the source code for your application, you may purchase a license from Oracle."(see http://www.oracle.com)

Commercial customers that use, modify and embed Berkeley DB into their own applications might be reluctant to use the OS version. These applications may be products per se or, more frequently, they are part of a more complex system that customers produce and sell. In both instances, it is clear that since customers want to keep proprietary control on their derived products, they may be willing to pay in order to be relieved from the reciprocal provision imposed by the open source version.

The software vendor benefits from releasing for free the open source version thanks to the contributions of OS adopters. These contributions, either "code" or "non-code", are then incorporated into the proprietary version and this helps to ameliorate product's quality.[3] It deserves to be noticed that typically the software vendor keeps strong control on the open source project and maintains the possibility of re-using "code" contributions by requiring external programmers to grant the permission to incorporate the lines of code that they have written into the proprietary version.[4]

As argued in the paper, the licensing terms of the open source version of the software are pivotal in the commercial sustainability of a dual licensing strategy. On the one side, a restrictive license, e.g. a license that imposes the reciprocal provision, represents an important safeguard against the possible cannibalization of the proprietary version of the software since it discourages some potential customers from adopting the open source version.[5] On the other side, the licensing terms affect also the size of the OS community, as well as the incentives that OS programmers have in contributing to the software project. As documented in many empirical studies, the terms of distribution of an OS project are an important determinant of its overall progress; Comino et al. (2007) have shown that OS projects released according to a more restrictive license are less likely to succeed. Others have shown that more restrictive licensing terms negatively affect the contribution (average lines of code written) of the members of the OS community (see and Fershtman and Gandal, 2007).

More specifically, in this paper we consider a profit maximizing firm that is developing a software project targeted to commercial customers. The firm either develops the project completely in-house or it employs a dual

[2] Oracle suggests that beyond relieving from the reciprocal provision, there are additional benefits of adopting the proprietary version: in the description of its dual licensing strategy, Oracle argues that the proprietary version of the software includes "legal assurances, warranties, and a wide array technical and aftersale services provided by a full-time dedicated development team". Furthermore, many OS software projects are distributed under licenses that allow the licensor to terminate the agreement conditional on the occurrence of specific events, and this clearly puts the customer at risk in case she/he needs to invest money and effort in using the software (see Rosen, 2004 for a discussion of the so-called "patent termination clauses").

[3] Sun Microsystems, the producer of MySQL describes its dual licensing strategy as follows: "We have over 4 thousand paying customers who have chosen the commercially-licensed MySQL server, and we have over 4 million users who use MySQL under the GNU General Public License (GPL). [...] Thanks to our commercial customers, we can afford to develop and improve the product at a fast pace. [...] And thanks to the huge user community, MySQL undergoes rigorous battle-testing"; see http://www.mysql.com/news-and-events/newsletter/2003-11/a0000000220.html.

[4] This is, for example, the case of the *Open source project submission agreement* by Digium, the well known producer of the telecommunications software Asterisk. MySQL, instead, follows a different approach: it rewrites and reassembles the lines of code written by OS programmers and then includes them in the proprietary version of the software (see Välimäki, 2005).

[5] Note that once the code has been released to the OS community, cannibalization may take the form of the so called *forking*: OS programmers might download the code and start independent development on it. To prevent this risk, the firm must maintain a strong leadership in the management of the project. In the theoretical model, we do not account explicitly for the risk of forking.

Please cite this article in press as: Comino, S., Manenti, F.M. Dual licensing in open source software markets. Inf. Econ. Policy (2011), doi:10.1016/j.infoecopol.2011.07.001

licensing strategy. In this latter case, it is crucial to manage appropriately the open source license in order to balance pros (the contributions of the OS community) and cons (the risk of cannibalizing the proprietary package) of "going OS". Assuming that customers have heterogenous preferences towards the restrictions imposed by the OS license, we derive the conditions under which dual licensing is profitable. Moreover, we discuss how an appropriate definition of the licensing terms allows the firm to optimally segment its potential customers into two groups: those who adopt the OS version, and that contribute to enhance the software quality, and those that pay for the proprietary version.

Our paper contributes to various strands of economic literature. As this introduction should have made evident, dual licensing represents a form of versioning; many authors have shown that versioning may be a profitable strategy when it allows the firm to enlarge significantly its market share or to benefit from increased consumption externalities (Shapiro and Varian, 1998; Belleflamme, 2005; Economides, 1996; Gayer and Shy, 2003). In our paper, the benefit of versioning is related to a "development externality": the OS community contributes to improve the quality of the software, and this allows the firm to charge a larger price for the proprietary version of its product.

The literature on platforms is also closely related to our paper (see Parker and Van Alstyne, 2005). Producers of personal computers, cell phones, or games often grant access to their systems in order to spur complementary innovations by downstream developers. As in our case, in choosing the degree of openness the platform owner trades off between two opposing effects: a more open platform is more effective in stimulating downstream innovation, but, at the same time, it may prevent the owner from obtaining adequate returns. With the relevant exception of Parker and Van Alstyne (2005), the literature on platforms is to a great extent empirical. The main difference with Parker and Van Alstyne (2005) is that we consider a setting where downstream developers are also users of the product; as stressed in von Hippel (2005), this is a distinctive feature of open source software. In our paper, the amount of "active users", i.e. those that adopt the OS version and contribute to improve the product, and the amount of "pure users", namely those that purchase the proprietary version and that do not contribute, is endogenously determined by the firm's licensing and pricing strategies.

The theoretical literature on the "economics of open source" has focussed mainly on the competition between open source and proprietary software; little has been done to achieve a better understanding of the rationales for commercial vendors to go open source (see Lanzi, 2009, for a recent review). An exception, closer to our paper, is represented by Mustonen (2005) where the author explicitly models a firm's decision to support a rival open source software project. The author shows that the firm may find it optimal to sustain OS when this promotes compatibility between the OS and the proprietary versions of the software, and when the OS community provides a sufficiently valuable development externality. It deserves to be noted that Mustonen's development externality differs substantially from ours'.

In Mustonen (2005) the strength of the externality is exogenous, while in our setting it is endogenously determined and proportional to the size of the open source community. More importantly, in Mustonen (2005), the development externality is relevant only in relation to the existence of the OS alternative, which exists independently of what the firm does; on the contrary, if rather than supporting an existing project, the firm "creates" the OS project, as in our model, then Mustonen's development externality would not have any impact on firms' behavior.

The rest of the paper is organized as follows: in Section 2 we present the model and we derive the main results, while in Section 3 we conclude. All the proofs that are not essential to understand the main arguments of the paper are in the appendix.

2. The model

Consider a commercial firm that has developed an embedded software. The software is directed to commercial customers, such as original equipment manufacturers, independent software vendors or value added resellers, that want to embed it into their own products.

The firm faces an alternative about how to commercialize the software. It can either distribute it only under proprietary licensing terms, or, alternatively, it can endorse a dual licensing strategy and distribute the software also under an open source license. In this case, the firm makes the open source version of the software available at no fee by posting it on a public repository.

When embedding the software, customers need to adapt it to their own products; typically, they improve the quality of the software by customizing it or, more simply, by adding new functionalities or by fixing existing bugs. Customers who adopt the open source version of the software (we refer to this mass of customers as the "open source community"), make these improvements publicly available; public availability implies that such improvements can be incorporated also into the proprietary version of the software, thus enhancing its quality. We refer to the contributions of the open source community as the development externality.[6]

Formally, we model the timing of the interaction between firm and customers in the following way:

1. the firm takes its distribution and pricing decisions. It decides whether to employ a dual licensing strategy (i.e. to release the open source version of the software together with the proprietary one) or to release the proprietary version of the software only. Afterwards, it sets the price and the licensing terms governing the versions of the software that are made available to customers. As we have clarified above, the open source version of the software (if released) is made available free of charge. The choice of the licensing terms amounts to define the degree of restrictions imposed on the possible uses

[6] The other main benefit of "going open source", namely the possibility of enlarging the installed base of users, is not considered in our framework. In fact, as shown in Lemma 1, in equilibrium, the market is always fully covered also when the firm does not release the open source version.

of the software; formally, the firm chooses $r_{os} \geqslant 0$: the larger r_{os}, the more restrictive the licensing terms of the OS version of the software. On the contrary, the proprietary version is sold at a positive price, denoted by p, while it is distributed without any restriction on its possible uses; formally the firm sets $r_p = 0$ for the proprietary version of the software. For the sake of simplicity, all through the paper firm's costs are normalized to zero;
2. customers, whose mass is normalized to *1*, take their adoption decision and then choose how much effort $e \geqslant 0$ to exert in order to improve the quality of the software; effort is costly and we denote this cost by $c(e)$.[7] OS adopters make their improvements publicly available, thus generating the development externality. In case a customer does not adopt any version of the software, then she enjoys a reservation utility u_o greater than zero.

The benefit that a customer obtains when adopting version $i = p, os$ of the software (proprietary or open source), depends on two factors: (*a*) a direct benefit $\pi(V,e,t,r_i)$ and (*b*) the benefit due to the development externality $\theta(r_{os})N$ accruing from the contributions of the OS community. We assume that the direct benefit $\pi(V,e,t,r_i)$ is positively affected by the quality of the software produced by the firm, $V > 0$, and by the effort exerted by the customer, e, while it decreases with the degree of restrictions imposed by the licence, r_i, and with the parameter t that measures how much such restrictions matter to the customer; in fact, since each customer embeds the software into her own product, a more restrictive license limits the possible uses and applications of the software and, consequently, it reduces the revenues that the embedder can earn by adopting it. We assume that customers are heterogeneous with respect to the effect of license restrictiveness; in particular, we assume that t is distributed according to the c.d.f. $F(t)$ over the support $[0,T)$, where $T > 0$ may be either finite or infinite.[8] Customers with a low t are little affected by license restrictiveness while customers characterized by a large t receive a strong disutility from r_i.

The development externality, $\theta(r_{os})N$, is increasing in N, the mass of open source adopters, and in the strength of the externality, $\theta(r_{os}) \geqslant 0$. Formally, $\theta(r_{os})$ is a function of the efforts exerted by the adopters belonging to the open source community when they customize and improve the software; in turn, as we clarify below, the efforts of the open source community are affected by the restrictions imposed by the OS license and this explains why we define the strength of the externality as a function of r_{os}.

In what follows, we assume that the overall benefit (direct benefit and development externality) from adopting version i of the software is:

$$B_i(V,e,t,r_i,r_{os}) = \begin{cases} \pi(V,e,t,r_i) + \theta(r_{os})N & \text{if } \pi(V,e,t,r_i) \geqslant 0 \\ 0 & \text{if } \pi(V,e,t,r_i) < 0. \end{cases} \quad (1)$$

The above expression shows that there is a weak form of complementarity between the direct benefit and the development externality: a customer benefits from the development externality only when the direct benefit is positive. This is a natural assumption; if the quality of the software released by the firm is so poor that the direct benefit from adoption is negative, then also the improvements provided by the open source community are worthless for the customer. For the sake of simplicity, in the definition of B_i, we assume that there are no other forms of complementarities between direct benefit and development externality and, therefore, the two terms $\pi(V,e,t,r_i)$ and $\theta(r_{os})N$ enter separately into the benefit function.

Following the above arguments, the net utility that a customer obtains from a software is B_i minus the cost of effort e, and, for the proprietary version, the price charged by the firm; formally:

$$U_{os}(V,e,t,r_{os},r_{os}) = B_{os}(V,e,t,r_{os},r_{os}) - c(e),$$
$$U_p(V,e,t,r_p,r_{os}) = B_p(V,e,t,r_p,r_{os}) - c(e) - p.$$

These two expressions show the basic trade-off that a customer faces when choosing between the two versions of the software: the proprietary version is available at price p but it is released without any restriction on the possible uses, $r_p = 0$, while the OS version can be obtained free of charge but it is distributed with restrictions $r_{os} \geqslant 0$.

Finally, notice that when the firm releases only the proprietary version, there is no open source community working to improve the software; in this case, the development externality is absent and those that purchase the software enjoy only the direct benefit from adoption.

2.1. Customers' choices

For the sake of simplicity, it is useful to proceed using specific functional forms for the various terms of the utility functions U_{os} and U_p; in particular, we assume that the cost function is quadratic, $c(e) = e^2/2$, and that the overall benefit from the adoption of the version i of the software is:

$$B_i(V,e,t,r_i,r_{os}) = \begin{cases} e\sqrt{2(V - tr_i)} + \theta(r_{os})N & \text{if } t \leqslant V/r_i \\ 0 & \text{if } t > V/r_i. \end{cases}$$

As already discussed above, a customer benefits from the development externality provided that the direct benefit is positive. This occurs when the restrictions imposed by the license are not too severe and they do not affect too much the overall quality of the software; formally, when $t \leqslant V/r_i$. On the contrary, when $t > V/r_i$ the customer does not obtain any benefit (neither the direct nor the development externality) by adopting the version i of the software.

[7] For simplicity, we assume that all customers improve the software. This assumption is taken with a little loss of generality: the model can be easily extended to the case where only a fraction of users also develop new functionalities; in this case, dual licensing would be profitable only when the mass of "users-developers" is sufficiently large.

[8] The level of t depends both on the nature and on the possible uses of the software. Since customers use the code as an input to produce other, derived, software that they either sell directly or that they embed into their own products, t is larger when the derived software represents the core of the customers' products/technologies: the more relevant the derived software in the embedded system, the larger the damage for the embedder if forced by the license to release the code under reciprocal licensing terms.

Please cite this article in press as: Comino, S., Manenti, F.M. Dual licensing in open source software markets. Inf. Econ. Policy (2011), doi:10.1016/j.infoecopol.2011.07.001

Once the adoption decision has been taken, a customer chooses how much effort to devote to improve the quality of the software; formally, she chooses e to maximize U_i, with $i = p, os$.[9] Simple algebra is enough to show that the optimal effort is:

$$e^*(t, r_i) = \begin{cases} \sqrt{2(V - tr_i)} & \text{if } t \leqslant V/r_i \\ 0 & \text{if } t > V/r_i. \end{cases} \quad (2)$$

This expression suggests some interesting observations. The optimal level of effort decreases with t and r_i: the more the customer is annoyed by the restrictions imposed by the license (the larger t), and the more restrictive the license (the larger r_i), the less the amount of effort she is induced to exert in improving the version i of the software that she has adopted. Therefore, those that adopt the proprietary version of the software exert the maximum amount of effort (remember that $r_p = 0$) whileOS adopters exert a (lower) level of effort which decreases with r_{os}.[10]

In what follows, we do not specify a functional form for the strength of the development externality $\theta(r_{os})$, nor we make assumptions on the c.d.f. $F(t)$. However, since the effort exerted by all those that adopt the OS version of the software decreases with r_{os}, we assume that $\theta(r_{os})$ is a decreasing function of license restrictiveness. Formally, in the rest of the paper we will assume that $\theta(r_{os})$ is differentiable and with derivative $\theta'(r_{os}) \leqslant 0$.

Plugging expression (2) into $U_p(V,e,t,r_p,r_{os})$ and $U_{os}(V,e,t,r_{os},r_{os})$, and recalling that $r_p = 0$, it is easy to derive the levels of net utility that a customer enjoys conditional on $e = e^*(t,r_i)$:

$$U_p^*(V, p, r_{os}) = V + \theta(r_{os})N - p \quad \text{and}$$

$$U_{os}^*(V, t, r_{os}) = \begin{cases} V + \theta(r_{os})N - tr_{os} & \text{if } t \leqslant V/r_{os} \\ 0 & \text{if } t > V/r_{os}. \end{cases} \quad (3)$$

2.2. Firm's strategy

The firm chooses whether to release only the proprietary version of the software or to employ a dual licensing strategy; afterwards, it takes its licensing and pricing decisions. As we have clarified above, the firm releases the proprietary version with no licensing restrictions, $r_p = 0$, while the open source version (if released) is made available to customers free of charge. Therefore, the firm chooses only the price p for the proprietary version, and, in case of dual licensing, it determines the degree of restrictions r_{os} imposed on the open source version of the software. As far as the initial decision (dual licensing or proprietary version only) is concerned, in principle we should compare the profits that the firm achieves in the two cases; however, the analysis can be limited to dual licensing, once noted that releasing the code to the OS community at an extremely restrictive license is equivalent to sell the proprietary version only. In fact, when the firm sets the degree of license restrictiveness to infinity, none would be willing to adopt the OS version and this makes, de facto, the firm distributing only the proprietary version of the software. The decision to release only the proprietary version is therefore encompassed into the optimal dual licensing strategy; formally, the choice $r_{os} \to \infty$ can be interpreted as the firm not making available the open source version of the software.

In what follows, we concentrate on the case where $V > u_o$; the alternative case $V \leqslant u_o$ is of no interest since the strategy of distributing only the proprietary version would never be profitable; in fact, without the benefit accruing from the development externality, whatever the price p, each customer would prefer not to buy the software.

Before providing a sufficient condition for dual licensing to be the most profitable alternative for the software vendor we provide some preliminary results about how the firm sets p and r_{os}. The first important result is characterized by the following lemma:

Lemma 1. *The firm sets p and r so that the market is fully covered.*

Proof. In order make positive profits, the firm must sell at least one copy of the proprietary version; a necessary condition for this to occur is $U_p^*(V, p, r_{os}) \geqslant u_o$. Since customers are heterogenous only with respect to license restrictiveness then, given that $r_p = 0$, they all obtain the same utility when adopting the proprietary version; therefore, the condition $U_p^*(V, p, r_{os}) \geqslant u_o$ is sufficient to ensure that all customers obtain at least their reservation utility when adopting the proprietary version. As a consequence, at the equilibrium, each customer adopts one version of the software, either the proprietary or the OS one. □

According to this Lemma, the firm's pricing and licensing choices are such that each customer adopts one of the two versions of software; in what follows, in order to emphasize that customers' adoption decisions depend on p and r_{os}, we denote the masses of OS and proprietary adopters as $N(p,r_{os})$ and $1 - N(p,r_{os})$, respectively.

The next lemma shows another condition that the pricing and licensing decisions of the firm must satisfy.

Lemma 2. *The firm sets p and r_{os} to extract perfectly the gross surplus of those who adopt the proprietary version of the software. Formally, $V + \theta(r_{os})N(p,r_{os}) - p = u_o$.*

Proof. See the appendix. □

The intuition for the above lemma is simple. As in a standard monopoly model with unit demand, the firm optimally sets the price and the restrictiveness of the OS license in order to extract all the surplus obtained by the customers who adopt the proprietary version of the software.

[9] Notice that, when choosing e, each OS adopter does not take into account that her choice affects $\theta(r_{os})$ and, consequently, the extent of the development externality that goes to the benefit of all other customers; hence, the optimal level of effort of OS adopters is lower than the efficient one. As it is typical in the presence of positive externalities, our analysis is characterized by under-provision of effort, although we do not model the case of a pure public good, as in Johnson (2002).

[10] This is consistent with the empirical literature on OS software, which finds evidence that the level of engagement of the OS community tends to decrease with the restrictions imposed by the license. See among others, Comino et al. (2007) and Fershtman and Gandal (2007).

Thanks to Lemma 2, the choice of the firm can be considerably simplified; condition $V + \theta(r_{os})N(p, r_{os}) - p = u_o$ implicitly defines the optimal price for the proprietary version of the software as a function of the degree of licence restrictiveness, r_{os}. Denoting this price as $p(r_{os})$, the firm's maximization problem can be written as follows:

$$\max_{r_{os} \geqslant 0} \quad \pi(r_{os}) = p(r_{os})(1 - N(p, r_{os})).$$

We are now in the position to state the main result of the paper.

Proposition 1. *When the strength of the development externality is sufficiently large, then it is optimal to employ a dual licensing strategy. Formally, when* $\lim_{r_{os}\to\infty} \theta(r_{os}) > V - u_o$, *the firm finds it optimal to set a finite* r_{os}.

Proof. See the appendix. □

This result is intuitive. The benefit of dual licensing accrues from the development externality: the contributions of the OS community improve the quality of the code and allow the firm to charge a larger price for the proprietary version. On the other hand, dual licensing entails the risk of cannibalizing the market since the open source version of the software competes with the proprietary one. Dual licensing is a profitable strategy for the firm, whenever the strength of the development externality, $\theta(r_{os})$, is large relative to the quality of the software that the firm is able to develop in-house, V. In particular, the proposition shows that when the strength of the externality is sufficiently large, the firm finds it optimal to set r_{os} at a sufficiently low level so that there exists a positive mass of open source adopters.

Proposition 1 provides a sufficient condition for dual licensing to be profitable for the firm. In order to proceed further and to better qualify the optimal dual licensing strategy actually employed by the firm, we need to make additional assumptions about the shape of the strength of the development externality. The next proposition characterizes the optimal strategy chosen by the software vendor when the strength of the externality takes a rather specific form, namely $\theta(r_{os})$ is constant and equal to θ.[11]

Proposition 2. *Suppose that* $\theta(r_{os}) = \theta$ *for all* r_{os}, *then the firm optimally employs a dual licensing strategy when* $\theta > V - u_o$. *Under dual licensing, the size of the OS community is equal to* $\frac{\theta - V + u_o}{2\theta}$, *regardless of the c.d.f.* $F(t)$.

Proof. See the appendix. □

As just discussed when commenting Proposition 1, the decision to employ a dual licensing strategy relies entirely on the comparison between the strength of the externality and the in-house quality of the code. Proposition 2 highlights an additional interesting feature of the equilibrium strategy: when $\theta(r_{os})$ is constant, the firm sets p and r_{os} in a way such that the mass of OS adopters is independent of the distribution of customers' preferences towards license restrictiveness.

In order to interpret this latter result, it is useful to consider the maximization problem in terms of N rather than r_{os}, provided that there is a one-to-one mapping between r_{os} and N. In this respect, Lemma 2 implicitly defines the optimal price as a function of N, formally $p(N) = V + \theta N - u_o$. The firm chooses N to maximize its profits $p(N)(1 - N)$. When θ is constant, an increase in N has two effects: on the one side, firms' profits go up by $\theta(1 - N)$; on the other side, a larger N, reduces sales and, therefore, profits of an amount equal to $p(N) = V + \theta N - u_o$. As shown in Proposition 2, these two effects cancel each other out when the mass of OS adopters is $(\theta - V + u_o)/(2\theta)$; this expression does not depend on the function $F(t)$, thus explaining why at the equilibrium the optimal size of the OS community is independent of the distribution of customers' preferences towards r_{os}.

Let us now consider the case with $\theta'(r_{os}) < 0$; in analogy with the findings of Proposition 2, one may wonder whether the firm sets the license restrictiveness and the price at levels $\tilde{r}_{os}$, and $\tilde{p}$ such that the size of the OS community is $(\theta(\tilde{r}_{os}) - V + u_o)/(2\theta(\tilde{r}_{os}))$. When the strength of the development externality is affected by the license restrictiveness, things become more articulated than before; in this scenario, an additional effect must be taken into account by the firm when deciding r_{os}: a less restrictive license is generally more desirable since it increases the strength of the development externality. We highlight these arguments in Proposition 3.

Proposition 3. *Suppose that* $\theta'(r_{os}) < 0$; *whenever the firm finds it optimal to endorse a dual licensing strategy, it does so by setting* $r_{os} < \tilde{r}_{os}$, *where* $\tilde{r}_{os}$ *satisfies the condition* $N = (\theta(\tilde{r}_{os}) - V + u_o)/(2\theta(\tilde{r}_{os}))$.

Proof. See the appendix. □

This proposition is interesting and suggests that when $\theta'(r_{os}) < 0$, the firm tends to endorse a more "pro OS" strategy by releasing the code to the community under less restrictive terms; this is a good thing for the firm since it stimulates the contributions from the OS developers, which translates into higher quality of the proprietary version and therefore larger profits.

We conclude this section with a final observation related to the welfare effects of dual licensing:

Remark 1. When the firm employs a dual licensing strategy, it induces a Pareto improvement.

Obviously, whenever the firm chooses to dual licence its code, it does so because it obtains larger profit. But also customers may be better off. When the firm releases only the proprietary version of the software it charges $p = V - u_o$; in this case, all individuals end up enjoying their

[11] The assumption that $\theta(r_{os}) = \theta$ for all r_{os} may seem at odds with the observation that $e^*(t, r_i)$ decreases with r_{os}. One way of reconciling these two facts is the following. The individual effort is, in part, customer specific (it goes to the benefit only of the customer who has exerted it) and, in part, it is of general interest (it goes to the benefit of other software adopters, giving rise to the development externality). In this scenario we may assume that only the customer specific part is (negatively) affected by the degree of licensing restrictiveness while the part of general interest does not change with r_{os}.

reservation utility u_o. With dual licensing, those that purchase the proprietary version still obtain u_o (see Lemma 2); however, customers adopting the OS version of the software obtain a utility which is strictly larger than the reservation level. The decision to provide the code to the OS community generates some value through the development externality; part of this value goes to the firm and part to individuals, thus explaining the Remark.

2.3. A specific example

In order to characterize fully the optimal strategy endorsed by the firm, we reconsider the case of constant strength of the externality; furthermore, let us assume that t is uniformly distributed over $(0, b)$.

Corollary 1. *When* $t \sim U(0,b)$ *and* $\theta(r_{os}) = \theta$, *with* $\theta \in (V - u_o, V + u_o)$ *then the firm optimally employs a dual licensing strategy with* $\bar{p} = \frac{V - u_o + \theta}{2}$, *and* $\bar{r}_{os} = \frac{\theta(V - u_o + \theta)}{b(\theta - V + u_o)}$. *License restrictiveness is such that* $\frac{\partial \bar{r}_{os}}{\partial V} > 0, \frac{\partial \bar{r}_{os}}{\partial b} < 0$, *and* $\frac{\partial \bar{r}_{os}}{\partial \theta} < 0$ *when* $\theta \in \left(V - u_o, \left(\sqrt{2} + 1\right)(V - u_o)\right)$, *while* $\frac{\partial \bar{r}_{os}}{\partial \theta} > 0$ *when* $\theta \in \left(\left(\sqrt{2} + 1\right)(V - u_o), V + u_o\right)$.

Proof. See the appendix. □

The positive relationship between $\bar{r}_{os}$ and V can be explained following the same arguments used to discuss Propositions 1, and 2: as V increases, the firm benefits from employing a more "proprietary strategy", i.e. by selecting a more restrictive license.

Consider now the role of b that parameterizes the distribution of customers preferences. From Proposition 2 we know that the firm sets the license restrictiveness in order to optimally segment customers into OS and proprietary adopters. When b gets larger the mass of customers that are substantially affected by the license restrictiveness increases; therefore, the firm needs to reduce r_{os} in order to enlarge the mass of OS adopters up to the optimal size defined in Proposition 2.

The impact of an increase in the strength of the development externality on $\bar{r}_{os}$ is more articulated and it entails to two opposite effects. A larger value of θ signals that the contribution of the OS community is highly valuable. Nonetheless, a larger θ makes the open source version of the software also a stronger competitor vis a vis the proprietary one; more specifically, as θ increases a larger share of customers is attracted by the OS version of the product. The former effect dominates whenever the size of the OS community is relatively small, that is when the strength of the externality is not too large, $\theta \in (V - u_o, \left(\sqrt{2} + 1\right)(V - u_o))$. In this case, the firm benefits from augmenting the size of the community through a reduction in the level of license restrictiveness. On the opposite, the "competition effect" prevails when the OS community is already sufficiently large; in this case, the firm reacts to a further increase in θ by selecting a larger $\bar{r}_{os}$.

3. Discussion and future research

In this paper we have proposed a theoretical model to study the characteristics and the commercial sustainability of a particular open source business strategy known as dual licensing. The focus is on the decision of a software vendor about whether to commercialize only the proprietary version of the software or to employ a dual licensing strategy, in a context where customers are commercial firms that are harmed by the restrictions imposed by OS licenses. We have shown that dual licensing is preferred when the feedbacks of the OS community (the development externality) are valuable enough compared to the quality of the software that the firm is able to develop on its own.

Our analysis points to the crucial role of OS licensing schemes for firms embracing open source strategies. Through an appropriate definition of the licensing terms of distribution of the OS version of the software, the firm balances the opposing effects of going open source. A more restrictive license protects the proprietary version of the software against the risk of cannibalization at the cost of reducing the size of the OS community that contributes to software development; moreover, licensing terms also affect OS programmers' incentives to contribute to improve the software.

Even though in presenting the model we have focussed on the role of reciprocal provisions in making dual licensing viable, our results have a broader interpretation. As discussed in the Introduction, there are additional dimensions of OS licenses that might disturb potential customers; in these cases, a software house may profitably go OS and sell an "upgraded" version of the software to those customers who are willing to pay to be freed from the specific provisions/limitations of the OS version.

In stressing the importance of an appropriate management of OS licenses for software vendors, we offer a possible explanation to one of the most debated phenomenon in the OS world, known as "license proliferation" (see Rosen, 2004). At June 2011, nearly 70 different licensing schemes have been registered as OS licenses; these licenses differ along several dimensions.[12] Interestingly, various commercial vendors have created their own open source license, thus confirming a possible strategic role in the "design" of the license.[13]

One simplifying assumption that we have implicitly made in the paper and that deserves further discussion is that the only way for the firm to benefit from the contribution of the community is by making the OS version of the code freely available on a public repository. This assumption is made on practical grounds; the fact that a vast majority of OS projects hosted on public repositories

[12] Take, for instance, the reciprocal provision; not all the OS licenses impose such provision (this is the case of the BSD and the other so-called "academic licenses") while, at the same time, an extreme heterogeneity in terms of the degree of reciprocity imposed on derivative works can be observed between those licenses that do have reciprocal provisions.

[13] The case of Nokia is emblematic. At the url `opensource.nokia.com/` several different software projects are available for download and often projects are licensed under different terms. Some projects are distributed under the Nokia Open Source License, others are available under different OS licensing templates such as: GPL, BSD, Mozilla Public License, LGPL, and others. Similarly, also IBM, Intel and Microsoft have created their OS license.

Please cite this article in press as: Comino, S., Manenti, F.M. Dual licensing in open source software markets. Inf. Econ. Policy (2011), doi:10.1016/j.infoecopol.2011.07.001

such as SourceForge.net and sponsored by commercial vendors are freely available, goes exactly in this direction. More specifically, this assumption is supported by the observation of the strategies adopted by those firms, such as Oracle and Sun Microsystems, whose experiences have been inspiring our paper. This assumption, however, is not innocuous since it implies that the firm cannot do better by following other more articulated strategies such as, for example, selling at a positive price the OS version.

Finally, in the paper we focus on the behavior of a monopolist producing a certain software. An important extension that we leave for future research relates to the role of open source strategies in a competitive framework.

Appendix A

Proof of Lemma 2. Consider a customer characterized by a certain disutility t from license restrictiveness. Given p and r_{os}, she adopts the OS version of the software provided that the following three conditions are jointly satisfied: (*i*) $t \leqslant p/r_{os}$, (*ii*) $t \leqslant (V+\theta(r_{os})N-u_o)/r_{os}$ and (*iii*) $t \leqslant V/r_{os}$. Condition (*i*) requires that the OS version is preferred to the proprietary one, $V+\theta(r_{os})N-tr \geqslant U_p^*$, condition (*ii*) that the OS version guarantees a utility larger than the reservation value, $V+\theta(r_{os})N-tr \geqslant u_o$, while condition (*iii*) implies that by adopting the OS version the customer enjoys a positive direct benefit. Moreover, notice that, in order to sell the proprietary version to some customers, the firm sets a price p such that $U_p^* \geqslant V+\theta(r_{os})N-tr$ for some $t\in[0,T)$; this fact implies that $p/r_{os} < T$.

From the previous observations it follows that the size of the OS community, as a function of p and r_{os}, is $N(p,r_{os}) = F(\min\{p/r_{os},(V+\theta(r_{os})N-u_o)/r_{os},V/r_{os}\})$.

In order to prove the lemma, we need to consider three cases depending on the value taken by $\min\{p/r_{os},(V+\theta(r_{os})N-u_o)/r_{os},V/r_{os}\}$.

Case 1: $\min\{p/r_{os},(V+\theta(r_{os})N-u_o)/r_{os}, V/r_{os}\} = p/r_{os}$.

Notice that the customer characterized by $t = p/r_{os}$ is indifferent between adopting the OS or the proprietary version of the software and that the net utility that she gets is exactly identical to the benefit received by all those who adopt the proprietary version. To prove the lemma, let us proceed by contradiction and suppose that the firm chooses p and r_{os} so that the proprietary adopters and the indifferent customer obtain a net utility strictly larger than u_o, formally $U_p^*(V,p,r_{os}) = V+\theta(r_{os})F(p/r_{os})-p = U_{os}^*(V,p/r_{os},r_{os}) = V+\theta(r_{os})F(p/r_{os}) - \frac{p}{r_{os}}r_{os} > u_o$. It is easy to show that the firm can do better by increasing marginally, and in the same proportion, p and r_{os} up, respectively, to $p' = p(1+\varepsilon)$ and $r'_{os} = r_{os}(1+\varepsilon)$, where $\varepsilon > 0$ is a negligible number. At the new pair (p', r'_{os}), customer $t = p/r_{os}$ is still indifferent between the two versions; moreover, given that ε is negligible, $U_p^*(V,p',r'_{os})$ and $U_{os}^*(V,p/r_{os},r'_{os})$ are still greater or equal than u_o and $\min\{p'/r'_{os},(V+\theta(r')N-u_o)/r'_{os},V/r'_{os}\} = p'/r'_{os} = p/r_{os}$. Therefore, the masses of customers adopting the two versions of the software do not change: $F(p/r_{os})$ is the mass of OS adopters, and $1-F(p/r_{os})$ is the mass of proprietary adopters. Nonetheless, since the firm is selling at a higher price it certainly makes larger profits, thus contradicting the initial assumption.

Case 2: $\min\{p/r_{os},(V+\theta(r_{os})N-u_o)/r_{os}, V/r_{os}\} = V/r_{os}$.

In this case, the mass of OS adopters is $F(V/r_{os})$ and the utility from purchasing the proprietary version is $U_p^*(V,p,r_{os}) = V+\theta(r_{os})F(V/r_{os})-p$. Suppose, by contradiction, that $V+\theta(r_{os})F(V/r_{os})-p > u_o$; this cannot be optimal since the firm may obtain larger profits by rising p marginally: a larger price does not affect neither the masses of OS and proprietary adopters nor $\theta(r_{os})$, therefore it ensures larger profits.

Case 3: $\min\{p/r_{os},(V+\theta(r_{os})N-u_o)/r_{os}, V/r_{os}\} = (V+\theta(r_{os})N-u_o)/r_{os}$.

In this case, the mass of OS adopters is $F((V+\theta(r_{os})N-u_o)/r_{os})$; the proof goes exactly in the same way as in Case 2. □

Proof of Proposition 1. From Lemma 2 we know that at the equilibrium $p = V+\theta(r_{os})N(p,r_{os})-u_o$, and this implies that the mass of OS adopters is:

$$N(p,r_{os}) = \min\{p/r_{os},(V+\theta(r_{os})N-u_o)/r_{os},V/r_{os}\} = \min\{p/r_{os},V/r_{os}\}.$$

Therefore, we need to distinguish two cases: (*i*) $\min\{V/r_{os},p/r_{os}\} = p/r_{os}$, and (*ii*) $\min\{V/r_{os},p/r_{os}\} = V/r_{os}$.

Case 1: $\min\{p/r_{os},V/r_{os}\} = p/r_{os}$.

In this case, the profits of the firm are equal to $\pi(r_{os}) = p(r_{os})(1-F(p/r_{os}))$, and, from Lemma 2, the price $p(r_{os})$ is implicitly defined by the condition $p = V+\theta(r_{os})F(p/r_{os})-u_o$. Consider the derivative of $\pi(r_{os})$ with respect to r_{os}; simple calculations show that $\pi'(r_{os}) = p'(r_{os})\left(1-F\left(\frac{p(r_{os})}{r_{os}}\right)\right) - p(r_{os})f\left(\frac{p(r_{os})}{r_{os}}\right)\frac{p'(r_{os})r_{os}-p(r_{os})}{r_{os}^2}$, where, $f(t) = \frac{dF(t)}{dt}$. Applying, the implicit function theorem to $p = V+\theta(r_{os})F(p/r_{os})-u_o$, it follows that:

$$p'(r_{os}) = \frac{\theta(r_{os})f\left(\frac{p}{r_{os}}\right)p - r_{os}^2\theta'(r_{os})F\left(\frac{p}{r_{os}}\right)}{r_{os}\left(\theta(r_{os})f\left(\frac{p}{r_{os}}\right)-r_{os}\right)}.$$

Using this expression and Lemma 2, the derivative $\pi'(r_{os})$ becomes (in order to avoid cumbersome notation, in what follows we omit the arguments of functions $\theta(r_{os})$, $\theta'(r_{os})$, $F(t)$, and $f(t)$):

$$\pi'(r_{os}) = \underbrace{f\frac{V-u_o+\theta F}{r_{os}}\frac{-\theta+2\theta F+V-u_o}{-\theta f+r_{os}}}_{(A)} + \underbrace{\frac{F(r_{os}(1-F)-f(V-u_o+\theta F))}{-\theta f+r_{os}}\theta'}_{(B)} \tag{4}$$

When $r_{os}\to\infty$, none of the customers adopts the OS version of the software, and then $\lim_{r_{os}\to\infty}\pi(r_{os}) = V-u_o$ which implies that $\lim_{r_{os}\to\infty}\pi'(r_{os}) = 0$. In order to prove the proposition, we need simply to show that $\pi(r_{os})$ converges to $V-u_o$ from above, namely that $\lim_{r_{os}\to\infty}\pi'(r_{os}) = 0^-$.

Please cite this article in press as: Comino, S., Manenti, F.M. Dual licensing in open source software markets. Inf. Econ. Policy (2011), doi:10.1016/j.infoecopol.2011.07.001

Consider term (B) of the above expression; note that as r_{os} goes to infinity, then, since $\theta' \leqslant 0$, this term is negative. Term (A) converges to 0^- provided that $\lim_{r_{os}\to\infty}\theta(r_{os}) > V - u_o$. This is enough to prove that there exist (at least) one finite value of r_{os} such that $\pi(r_{os}) > V - u_o$.

Case 2: $\min\{p/r_{os}, V/r_{os}\} = V/r_{os}$.

In this case, firm's profits are $\pi(r_{os}) = p(r_{os})(1 - F(V/r_{os}))$, and, from Lemma 2, the optimal price is $p(r_{os}) = V + \theta(r_{os})F(V/r_{os}) - u_o$, with $p'(r_{os}) = \theta'(r_{os})F(V/r_{os}) - \theta(r_{os})f(V/r_{os})V/r_{os}^2$. Consider the derivative of $\pi(r_{os})$ with respect to r_{os}; simple calculations show that $\pi'(r_{os}) = p'(r_{os})\left(1 - F\left(\frac{V}{r_{os}}\right)\right) + p(r_{os})f\left(\frac{V}{r_{os}}\right)$. Using the expressions of $p(r_{os})$ and $p'(r_{os})$, then $\pi'(r_{os})$ reduces to:

$$\pi'(r_{os}) = \underbrace{\frac{fV}{r_{os}^2}(-\theta + 2\theta F + V - u_0)}_{(A)} + \underbrace{F(1-F)\theta'}_{(B)}.$$

As in Case 1, when $r_{os} \to \infty$: (i) $\lim_{r_{os}\to\infty}\pi'(r_{os}) = 0$, (ii) the term (B) tends to zero from negative numbers, and (iii) the term (A) converges to 0^- provided that $\lim_{r_{os}\to\infty}\theta(r_{os}) > V - u_o$. □

Proof of Proposition 2. Following Proposition 1 two cases must be considered.

Case 1: $\min\{p/r_{os}, V/r_{os}\} = p/r_{os}$.

When $\theta(r_{os}) = \theta$, expression (4) reduces to:

$$\pi'(r_{os}) = f\frac{V - u_o + \theta F}{r_{os}}\frac{-\theta + 2\theta F + V - u_o}{-\theta f + r_{os}}.$$

The first order condition $\pi'(r_{os}) = 0$ is uniquely solved when $F = \frac{1}{2}\frac{\theta - V + u_o}{\theta}$. This implies that when $\theta \leqslant V - u_o$, the optimal size of the open source community is zero, while it is equal to $\frac{1}{2}\frac{\theta - V + u_o}{\theta}$ when $\theta > V - u_o$. Notice that when $\theta > V - u_o$, $\frac{1}{2}\frac{\theta - V + u_o}{\theta}$ identifies a maximum provided that $\pi(0) = 0$ and $\lim_{r_{os}\to\infty}\pi(r_{os}) = V - u_o$.

Case 2: $\min\{p/r_{os}, V/r_{os}\} = V/r_{os}$.

When $\theta(r_{os}) = \theta$, the first order condition becomes $\pi'(r_{os}) = fV(-\theta + 2\theta F + V - u_0)/r_{os}^2 = 0$ and the same arguments as in the previous case apply. □

Proof of Proposition 3. As in Proposition 1, Cases 1 and 2 must be considered; when $N = (\theta(\tilde{r}_{os}) - V + u_o)/(2\theta(\tilde{r}_{os}))$, then for both cases $\pi'(r_{os})$ becomes:

$$\frac{\left(\theta(\tilde{r}_{os})^2 - (V - u_o)^2\right)}{4\theta(\tilde{r}_{os})^2}\theta'(\tilde{r}_{os}).$$

It is immediate to see that this expression is negative. In fact: (i) $\theta'(r_{os}) < 0$ by assumption, and (ii) $\theta(\tilde{r}_{os}) > V - u_o$, since $\lim_{r_{os}\to\infty}\theta(r_{os}) > V - u_o$ (i.e. dual licensing is profitable), and $\theta'(r_{os}) < 0$. The firm prefers to set a lower r_{os} compared to $\tilde{r}_{os}$ and it is enough to complete the proof. □

Proof of Corollary 1. Lemma 2 and Proposition 2 provide the two conditions that the optimal p and r_{os} must satisfy: $V + \theta F(\min\{p/r_{os}, V/r_{os}\}) - p = u_o$ and $F(\min\{p/r_{os}, V/r_{os}\}) = (\theta - V + u_o)/(2\theta)$. Assuming that $\min\{p/r_{os}, V/r_{os}\} = p/r_{os}$, and using the assumption of uniform distribution, it follows that the optimal pricing and licensing strategy are $\bar{p} = (V - u_o + \theta)/2$ and $\bar{r}_{os} = [\theta(V - u_o + \theta)]/[b(\theta - V + u_o)]$; it is easy to check that $\min\{\bar{p}/\bar{r}_{os}, V/\bar{r}_{os}\} = \bar{p}/\bar{r}_{os}$ when $\theta < V + u_o$. The comparative statics is obtained by simply differentiating $\bar{r}_{os}$. □

References

Belleflamme, P., 2005. Versioning in the information economy: theory and applications. CESifo Economic Studies 51, 329–358.

Comino, S., Manenti, F.M., Parisi, M.L., 2007. From planning to mature: on the success of open source projects. Research Policy 36, 1575–1586.

Daffara, C., 2009. FLOSSMETRICS: the SME guide to open source software. <http://www.guide.flossmetrics.org/smeguide.pdf>.

Economides, N., 1996. Network externalities, complementarities, and invitations to enter. European Journal of Political Economy 12, 211–233.

Fershtman, C., Gandal, N., 2007. Open source software: motivation and restrictive licensing. International Economics and Economic Policy 4 (2), 209–225.

Gayer, A., Shy, O., 2003. Internet and peer-to-peer distributions in markets for digital products. Economic Letters 81, 51–57.

Johnson, J.P., 2002. Open source software: private provision of a public good. Journal of Economics & Management Strategy 11 (4), 637–662.

Jullien, N., 2006. New Economic Models, New Software Industry Economy. RNTL Report.

Lanzi, D., 2009. Competition and open source with perfect software compatibility. Information Economics and Policy 21, 192–200.

Moody, G., 2006. Does dual licensing threaten free software? Linux Journal, 27th July.

Mueller, M., 2007. OpenOffice.org Projects by Members. blogs.sun.com/GullFOSS/entry/openoffice_org_projects_by_members.

Mustonen, M., 2005. When does a firm support substitute open source programming? Journal of Economics & Management Strategy 14 (1), 121–139.

Parker, G., Van Alstyne, M.W., 2005. Innovation through Optimal Licensing in Free Markets and Free Software. SSRN eLibrary.

Rosen, L., 2004. Open Source Licensing. Software Freedom and Intellectual Property Law. Prentice Hall.

Seigo, A., 2006. The quest for project identity and definition. In: Keynote Speech Akademy Conference.

Shapiro, C., Varian, H., 1998. Versioning: the smart way to sell information. Harvard Business Review 76 (6), 106–114.

Välimäki, M., 2005. The Rise of Open Source Licensing: A Challenge to the Use of Intellectual Property in the Software Industry. Turre Publishing.

von Hippel, E., 2005. Democratizing Innovation. The MIT Press, Cambridge Mass, London.

Please cite this article in press as: Comino, S., Manenti, F.M. Dual licensing in open source software markets. Inf. Econ. Policy (2011), doi:10.1016/j.infoecopol.2011.07.001