UNIVERSITY OF CALIFORNIA, BERKELEY
GOLDMAN SCHOOL OF PUBLIC POLICY

GSPP10-001

The New (Commercial) Open Source: Does It Really Improve Social Welfare?

By
Sebastian von Engelhardt
Department of Economics and Business Administration
Friedrich-Schiller-Universit t, Jena

and

Stephen M. Maurer
Goldman School of Public Policy
and Berkeley Law School
University of California, Berkeley

This version: September, 2010

This paper is preliminary.
Before citing, please check
the authors' web sites
for upated versions.

Abstract

The number of open source ("OS") software projects has grown exponentially for at least a decade. Unlike early open source projects, much of this growth has been funded by commercial firms that expect to earn a profit on their investment. Typically, firms do this by selling bundles that contain both OS software and proprietary goods (e.g. cell phones, applications programs) and services (custom software). We present a general two-stage Cournot model in which arbitrary numbers of competing OS and closed source ("CS") firms decide how much software to create in stage one and how many bundles to supply in stage two. We find that the amount of OS software delivered depends on (a) the degree of substitutability between proprietary products, (b) the number of OS and CS firms competing in the market, and (c) the savings available to OS firms from cost-sharing. However, code-sharing also guarantees that no OS firm can offer better software than any other OS firm. This suppresses quality competition between OS firms and restricts their output much as an agreement to suppress competition on quality would. Competition from CS firms weakens this quality-cartel effect, thus mixed industries often offer higher welfare. We find that Pure-OS (Pure-CS) markets are sometimes stable against CS (OS) entry so that the required OS/CS state never occurs. Even where mixed OS/CS industries do exist, moreover, the proportion of OS firms needed to stabilize the market against entry is almost always much larger than the target ratio required to optimize welfare. We examine various policy options for addressing this imbalance with tax policy, funding of OS development, and procurement preferences. We find that the first-best solution in our model is to tax OS firms and grant tax breaks to CS firms. Conversely, government policies that fund OS development or establish procurement preferences for OS software actually increase the gap between desired and actual OS/CS ratios still further. Despite this, funding OS development can still improve welfare by boosting total (private + government) OS investment above the levels that a private cartel would deliver.

Keywords: open source; commercial open source; Cournot; procurement preferences

JEL: H25, L17, O34, O38

1. Introduction*

Despite differences in detail, the number of conceptually distinct incentives (e.g. patents, prizes, grants, contract research) that society uses to promote innovation is remarkably small (see Scotchmer 2004). Against this background, the emergence of fundamentally new "open source" (OS) methods for producing software[1] in the 1990s surprised and delighted observers. Furthermore, OS seemed to avoid proprietary or "closed source" (CS) software's worse feature—charging consumers a royalty for information that could theoretically be distributed at zero cost. This made it natural to ask whether OS could drastically improve welfare compared to CS. At first, this was only an intuition. Early explanations of OS were either ad hoc ("altruism") or downright mysterious (e.g. a post-modern "gift economy", see Raymond 1998). Absent a clear model of OS, no one could really be certain how much software the new incentive could deliver, let alone whether social welfare would best be served by OS, CS, or some mix of the two.

The past decade has seen considerable progress. Following Lerner and Tirole (2002)'s seminal article, economists showed that real world OS collaborations rely on many different incentives such as education, signaling, and reputation. Furthermore, they constructed detailed mathematical models of each mechanism (Maurer and Scotchmer 2004). The problem for policymakers was that the new models—despite a family resemblance—were all different and sometimes yielded contradictory insights. Worse, their predictive power was limited. Saying that certain OS projects were driven by a de-

*We are grateful to Suzanne Scotchmer, Andreas Freytag, Maria Alessandra Rossi, the participants of the "Augustin Cournot Doctoral Days" (Strasbourg, 1-3 April 2009) and the "Open Source, Innovation, and Entrepreneurship" (Jena, 14 January 2010) workshops for their comments on this paper. Sebastian von Engelhardt gratefully acknowledges financial support from the Klaus Tschira Foundation. Stephen Maurer gratefully acknowledges financial support from the Alfred E. Sloan Foundation and the U.S. National Science Foundation (Grant No. SES 20082737). The authors are solely responsible for any errors and oversights.

[1] OS represents a new intellectual property paradigm. OS software is marked by free access to the source code and is developed in a public, collaborative manner by non-paid volunteers as well as profit-seeking firms.

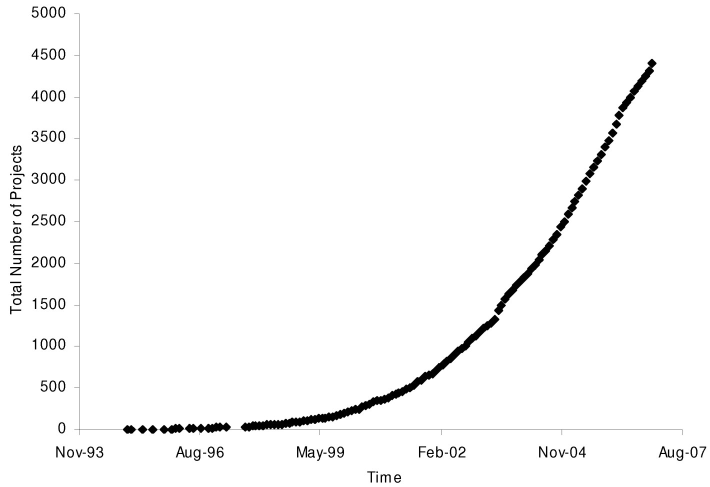
sire for reputation was one thing. Saying how much desire actually existed, let alone how much software it would generate, was another. Furthermore, the existence of multiple, competing models was disabling for policymakers. Government interventions that made sense for one set of OS incentives were likely to be irrelevant or even counterproductive for others.

The situation today is much improved. The reason is that the OS phenomenon has itself become more uniform. In a recent article, Deshpande and Riehle (2008) use automated web searches to compile a worldwide census of OS projects (Figure 1). Within their representative sample they report exponential growth from about 500 projects in 2001 to 4,500 in 2007. Strikingly, much of this growth appears to be commercially motivated, i.e. depends on substantial backing from corporations that expect to see a dollars-and-cents return on their investment.[2] Within this category, the great majority of projects involve a shared code base that no single company owns or controls. Each company then extracts benefits by selling complementary products that use this shared code. In practice, these products are very diverse and can include physical hardware, software and services (Riehle, personal communication).

This paper presents an analysis of why firms contribute to commercial OS collaborations, the welfare implications of these decisions, and possible government options for improving these outcomes. In particular, we explore the first general differentiated Cournot model in which firms invest in shared OS or private CS code to increase the quality (and profitability) of bundles containing a proprietary good. Like earlier papers, we find that the amount of OS produced reflects a balance between OS firms' ability to share costs and CS firms' greater ability to appropriate benefits to consumers. Depending on this balance, Pure-OS industries may or may not be welfare-superior to Pure-CS industries while Mixed-OS/CS industries are almost always welfare-superior to both. Unlike earlier contributions, our general model lets us systematically explore these competing effects for arbitrary combinations of substitutability

[2] Survey research has similarly detected a secular shift in OS motivations away from hobbyist-based production to a culture dominated by paid professionals working on company time (Ghosh et al. 2002a,b).

Figure 1: Total Number of Open Source Projects



(Source: Deshpande and Riehle 2008)

between competing proprietary products, and numbers of competing OS and CS firms.

Crucially, we find that Pure-OS industries hardly ever realize the full welfare benefits of cost sharing. The reason is that OS firms (unlike CS firms) share all software so that no firm can offer better quality than any other firm. This leads to a quality-cartel effect not seen in earlier models that suppresses OS code production. Paradoxically, then, we find that OS firms deliver relatively modest welfare gains unless and until CS firms are present in sufficient numbers to enforce quality competition. We also use our model to calculate which OS:CS firm ratios are stable against entry. We find that (a) many Pure-OS and Pure-CS markets are stable so that welfare-improving mixed OS/CS markets never arise, and (b) stable OS/CS mixed markets hardly ever contain enough CS firms to enforce optimal quality competition.

Understanding these effects matters. Most national governments already purchase large amounts of OS software and this spending is often explicitly justified as an attempt to promote OS over traditional CS code production models. We show below that such interventions can actually *decrease* social welfare. More generally, we use our model to evaluate welfare effects for a wide spectrum of possible interventions including taxation, direct funding of OS-development, and government procurement preferences.

The balance of this paper proceeds as follows. Section 2 sets the stage by describing some contemporary commercial OS projects and how various governments support OS, and reviewing previous efforts to model commercial OS. Section 3 presents our basic model and uses it to calculate the profit-maximizing output for arbitrary numbers of firms operating in Pure-CS, Pure-OS, and mixed OS/CS industries, and then shows how specific output decisions translate—through costs incurred and quality delivered to consumers—into net welfare. Section 4 identifies the conditions under which industries endowed with an initial mix of OS and CS firms are stable against entry. Crucially, it finds that few, if any of these equilibria are welfare-optimal. Section 5 examines several strategies that government could potentially use to intervene in these markets. Section 6 discusses the generality of these results and the extent to which more complicated models could lead to different conclusions. Section 7 presents a brief conclusion.

2. Background and Literature

2.1. The New (Commercial) Open Source.

Almost all of today's high tech products are computerized. While this is most obviously true for applications software (e.g. games), the point increasingly extends to hardware like cell phones and DVD players. In these industries, a product's quality—and hence consumer appeal—depends sensitively on the software it contains. Before the 1990s, companies usually wrote CS software for their products in-house. Since then, however, companies have increasingly turned to shared OS code instead. The number of companies in these OS communities typically ranges from a few dozen to many thousands. The following examples provide some idea of this range:

Small Communities. Most web servers are driven by an OS ”Lamp Stack” software suite that includes a Linux operating system, Apache web server software, MySQL database, and PHP/Perl/Python programming languages. Development is supported by a relatively small number of corporations like Novell, IBM, Oracle, and Borland who then bundle Lamp with their proprietary hardware and software. Small web developers also use Lamp in their businesses and contribute code back to the project.

Mid-Sized Communities. IBM opened its formerly CS Eclipse development tool in 2001 and created an independent foundation to manage further OS development in January 2004. 115 companies had joined the foundation as of 2006. Firms contribute code and are allowed to develop and sell proprietary applications programs ("plug-in tools") in return. In 2007, IBM's contribution fell below 50% of the total new additions to the code base (Mike Milinkovich,[3] personal communication)

Large Communities. Thousands of programmers developed the Linux code base so that they could develop custom software solutions for clients. More recently, Red Hat and other companies have begun selling various software products including support services, middleware, "Enterprise class" premium versions, and ready-to-install bundles ("distributions") based on Linux's OS modules. Furthermore, various firms like Nokia, Philips, or Sony, use Linux as embedded software for their products.[4]

Unlike traditional joint venture partners, OS collaborators have no formal obligation to contribute any particular level of effort to these projects. In-

[3]Mike Milinkovich is executive director of the Eclipse Foundation, Inc.

[4]Examples are Amazon's Kindle, Cisco's MDS and Nexus data switches, Linksys' WRT54G W-LAN router, various Motorola, Nokia, and Panasonic mobile phones, Philips' LPC3180 microcontroller, TomTom's GPS navigation systems, and various LG Panasonic, Samsung, and Sony LCD and plasma televisions. The most recent example of embedded OS is Android, a Linux-based software stack (operating system, middleware and key applications) for mobile devices. Acer, Barnes & Noble, Dell, HTC Corporation/Google, Lenovo, LG, Motorola, Samsung, and Sony Ericsson all manufacture and sell products that come preinstalled with Android.

stead, companies must continuously balance the cost-savings from shared code development against the risk that they will make their competitors' products more desirable. In what follows, we model these decisions as an extension of the familiar Cournot problem of how much output to produce.

2.2. Government Interventions

Governments are plainly intrigued with OS and have repeatedly flirted with various schemes to promote it. Probably the most comprehensive survey of existing and proposed initiatives is found in CSIS (2008). It reports that governments have experimented with a wide variety of incentives to promote OS companies. These include:

Procurement Preferences. Governments purchase large amounts of software and can potentially use these purchases to promote OS over CS and vice versa. At least sixteen countries have considered mandatory polices that would require government agencies and/or state-owned companies to purchase OS solutions whenever possible.[5] Softer versions of these proposals speak of ”preferences” for OS when its performance is comparable to CS. To date, at least ten national governments have adopted some version of these proposals[6] along with many state and local governments. High government adoption rates of OS in still other countries (e.g. France) suggests that unofficial preferences also exist.

Tax Incentives. Singapore offers tax breaks to firms that use Linux operating systems.

Government Funding. Hong Kong offers funding for companies that adopt or use OS. Israel offers grants of up to $100,000 to start-up companies

[5] Argentina, Belgium, Brazil, Bulgaria, Chile, Colombia, Israel, Italy, Netherlands, Ukraine, Finland, Portugal, Peru, and Venezuela.

[6] Australia, Belgium, Brazil, China, Malyasia, the Netherlands, Peru, South Africa, Spain, and Venezuela. Conversely, the UK, Canada, Germany and Slovenia choose between OS and CS solely on technical merits.

that use and develop OS. Governments have also funded a variety of institutes, projects, and private-public collaborations to develop[7] and facilitate user adoption[8] of OS software.

Grants Policy. Government research grants to academia and industry frequently require dissemination plans when software is produced. OS is by far the easiest way to meet these obligations. More formally, the United Kingdom has adopted a "default position" that government-funded software should be released under OS licenses (CSIS 2008).

While scholars have occasionally explored the case for such interventions (e.g. Schmidt and Schnitzer 2003), their welfare analyses have generally focused on the impact of government policy on OS collaborations driven by altruism, reputation, signaling, and other traditional incentives. Not surprisingly, these studies usually assumed that government spending could do little to influence OS code production. This assumption clearly needs to be revisited in an era when commercial incentives dominate OS production. Similarly, government policy in recent years has increasingly evolved from simple OS-promotion schemes to "a search for business models that can profitably blend open and proprietary processes and products" (CSIS 2008). However, earlier articles say relatively little about how the two sectors interact or how these interactions can be managed to improve welfare. We fill this gap by exploring how various government interventions including taxation, funding, and purchasing preferences influence output (and indirectly, welfare) for a very general commercial models in which OS and CS firms interact with one another.

2.3. Literature

Several articles have previously presented models analyzing how much for-profit firms invest in OS when they expect to sell a proprietary complement.

[7] China, Finland, Japan, South Korea, France, India, Slovakia, Spain, Thailand, Venezuela, Vietnam

[8] China, Czechoslovakia, Israel, South Africa, Taiwan, Cambodia, South Korea, Japan, Netherlands, Philippines, Thailand, Vietnam.

Most of these contributions are limited to the special case of a duopoly.

2.3.1. Duopoly Models

Baake and Wichmann (2004) consider a Bertrand model in which two firms decide about the quality of their software and whether to publish some parts as OS. Firms' costs rise with quality while OS reduces costs. OS facilitates further entry while quality impedes it. In equilibrium, both firms publish OS code, which yields higher quality either because of reduced costs or because of the threat of entry. The latter effect is even stronger and dominates when the software-products are strong substitutes. Though intriguing, these results are limited to the duopoly case, and to software-firms only. Baake and Wichmann also make the special assumption that CS costs rise faster than OS costs. Verani (2006) presents a Bertrand-duopoly model in which firms invest in either OS or CS software and then build products that use it. She finds that firms invest more in software when their products are substitutes and when OS software is used.

Henkel (2006) uses a duopoly model to explore the case of embedded Linux. Crucially, he assumes that all technologies are developed in-house without shared production of any kind; firms can, however, share costs by disclosing completed technologies to one another. Given this set-up, Henkel finds that each OS firm concentrates on developing whichever technology is most valuable to its business and copies the other technology from its rival. This creates a dynamic in which each company specializes in and controls the technology it values most so that total industry technology spending is biased upward. Henkel finds that OS industries deliver more technology and higher profits provided that firms do not compete too strongly with one another. However, these advantages disappear where both firms' products receive the same quality-increment from each technology. In this case, OS firms are reluctant to make their competitors stronger and therefore invest less than CS firms. Furthermore, firms are most likely to choose OS when competition is low, and each firm's technology needs are different.[9]

[9]Interestingly, Henkel also finds that firms choose OS where technology needs are similar and competition is high. However, he points out that this situation only occurs in the

2.3.2. Beyond Duopoly Models

Recently, some authors working independently of us have gone past simple duopoly models to explore firm decisions to invest in OS or CS. Llanes and de Elejalde (2009) and Casadesus-Masanell and Llanes (2009) provide models in the tradition of Hotelling's model. Both models feature a continuum of consumers who value the available products different, i.e. have heterogeneous tastes. Furthermore, each consumer buys just one package (bundle) or nothing.

In Casadesus-Masanell and Llanes (2009) consumers use software and a complementary service. The software is further segmented into a core program which can be used as a free-standing unit, and extensions which are valueless without the core unit. Casadesus-Masanell and Llanes examine how firms decide whether to develop one or both software components as OS or CS for the case of a monopoly, firm vs. non-profit OS project, and duopoly. They find, inter alia, that firms are more willing to open modules when consumer demand for the complementary good is strong, and the quality of OS software is boosted by exogenous user innovation at no cost.

Llanes and de Elejalde (2009) similarly consider a model in which each firm sells packages consisting of a primary good (which can be OS or CS) and a complementary private good. Consumers have idiosyncratic preferences so that they usually favor one firm's private good over others. However, rival firms can overcome this preference by investing in a technology that simultaneously increases the quality of both the primary good and also the complement. Llanes and DeElejalde present a two stage model in which a predetermined number of firms decide whether to produce OS or CS in the primary good, and then simultaneously decide the quality/price of the bundle that they will offer to consumers. They find that when most of the bundle's value comes from the primary good OS firms find it hard to appropriate profits from their investment in an open complement. This leads to outcomes in which a small number of firms choose CS and capture most of the market by delivering high quality code; the other firms become OS and deliver compara-

(presumably rare) case where competition is so strong that a CS duopoly would earn negative profits.

tively low quality code at a low price. However, this situation changes where consumers value the complement roughly as much as the primary. In this case, the cost advantage of code-sharing dominates so that all firms choose to become OS even though a hypothetical CS firm would produce higher quality software. This (theoretical) CS quality advantage reflects OS firms' limited ability to recover quality gains from consumers. The advantage disappears when most of the bundle's value comes from the complementary good.

2.3.3. Entry

Except for Baake and Wichmann (2004), Schmidtke (2006) is the only author known to us who systematically explores the effects of entry. Schmidtke analyzes OS business models in a non-differentiated Cournot oligopoly. Firms produce a homogeneous private good (e.g. a computer server) and invest in the quality of a homogeneous public good (OS software). He finds that increasing the number of firms in the market promotes welfare. The effects on each firm's output, prices, and profits depend on the slope of the marginal costs of software development.

3. A Simple Commercial OS Model: Calculating Output and Welfare

This section presents our very general Cournot model of how firms decide how much OS and CS software to produce and the welfare implications of those decisions. The model is based on von Engelhardt (2010) and combines aspects of (non-)cooperative R&D models (d'Aspremont and Jacquemin 1988) with the theory of differentiated oligopoly proposed by Dixit (1979) and further developed by, e.g. Singh and Vives (1984) and H ckner (2000). The model's main features include (a) separating firms' quality and production decisions into two separate and sequential phases, (b) a simple demand model in which consumer utility depends on the quality and quantity of bundles offered and quality can be summarized by single index variable, and (c) assuming that all quality-enhancing research can be performed either in OS or CS mode, i.e. that no technology is inherently private and un-shareable. We return to these design choices and the likely consequences of relaxing them in Section 6 ("Discussion") below.

3.1. Basic Model

We model private sector decisions to develop OS and CS software as a two-stage game:

In Stage One, profit-maximizing firms decide on how much code to develop (either as shared OS or private CS).

In Stage Two, the firms produce, for example, a DVD player or computer game whose performance depends on the code. They then sell the bundled products in markets that include one or more competitors.

For this paper, we stylize our model so that all quality is entirely attributable to software. This allows us to identify firms' *stage one* decisions with *quality* and their *stage two* decisions with *quantity*. As usual, we start by analyzing stage two and work backwards to stage one:

3.1.1. Stage Two: Decision on Quantity

Since the products of stages one and two are complements, consumers are not shopping for individual products but for bundles. Here each firm chooses whichever level of output q_i maximizes its profits $\pi_i = p_i q_i - c_i - C$ ("Cournot competition"), where p is price, q is the number of bundles sold, and C are the fixed costs of developing the stage two product[10] (c_i denotes the costs of software development determined in stage one). We model the price that consumers are willing to pay for each company's bundle by the inverse demand

[10]Without loss of generality, marginal costs of producing stage two products are normalized to zero.

function[11]

$$p_i = \alpha_i - q_i - \gamma \sum_{j \neq i} q_j \tag{1}$$

Here, the quality of Firm i's bundle is given by $\alpha_i = 1 + x_i$ where x_i is the amount of stage one software included in the product.[12] Finally, we use the variable γ to capture horizontal product differentiation, i.e. the extent to which manufacturers' stage two products compete with one another. This can range from cases of perfect substitutes where $\gamma = 1$ ("mobile phones vs. mobile phones") to $\gamma = 0$ scenarios where products hardly compete at all ("mobile phones vs. software-driven toasters").

Our inverse demand function is particularly convenient for analyzing Cournot competition because it leads to equilibria in which prices equal quantities. Thus stage two competition induces each firm to supply the following quantity of goods:

$$p_i = q_i = \frac{\alpha_i + \frac{\gamma}{(2-\gamma)} \sum_{j \neq i} \left(\alpha_i - \alpha_j\right)}{2 + \gamma(n-1)} = \frac{\alpha_i + \theta \sum_{j \neq i} \left(\alpha_i - \alpha_j\right)}{h}. \tag{2}$$

Please note that (2) has two important features. First, prices and quantities depend on the quality of firm i's bundle α_i and on the quality-difference regarding its competitors' bundles. This latter is weighted by θ, which indicates the degree to which firms compete on *quality* (stage one) as $\theta = \gamma/(2-\gamma)$ is a convenient rescaling of our original measure of substitution γ. Second, p_i and q_i decline when $h = 2 + (n-1)\gamma$ increase. Here h—which depends on the number of competitors (n-1) weighted by degree of substitution γ—captures competition without respect to quality differences. Hence, we will refer to h

[11] This is derived from a utility function used by Dixit (1979) and H ckner (2000). The literature also discusses a second type of demand function. However this would yield similar results, see von Engelhardt (2010).

[12] A more general formulation would be $\alpha_i = \beta + x_i$, with β as the quality of the 'stage two'-product. For the sake of simplicity we have normalized β to one. Values different than $\beta = 1$ do not change the results qualitatively. However, for $\beta > 1$ OS becomes more attractive.

as indexing the intensity of competition on *quantity*.

3.1.2. Stage One: Decision on Quality

In stage one firms decide how much code to develop and thus, implicitly the quality of their stage two bundles. Our quality equation $\alpha_i = 1 + x_i$, says that increased production of stage one software (x_i) makes the stage two bundle more valuable. On the other hand, no amount of software can increase a bundle's utility to infinity. We model this by defining an arbitrary upper limit (the "cutoff") beyond which software production has no further impact on quality.

In line with the literature, we assume that stage one software development can be approximated by a cost function with increasing marginal costs. This codifies the usual intuition that production encounters diminishing returns and is also mathematically necessary to suppress infinite code production. For simplicity we assume that this function is quadratic so that the cost of CS code is given by $c = \frac{1}{2}\phi x^{\text{cs}2}$ where ϕ is the slope of an increasing marginal cost function. Similarly, we model the total cost of OS-code as $c = \frac{1}{2}\phi X^{\text{os}2}$, with $X^{\text{os}} = \sum x_i^{\text{os}}$. This yields $c_i = \frac{1}{2}\phi x_i^{\text{os}} X^{\text{os}}$ for every OS firm, reflecting the cost-sharing across participating firms.[13] Both functions reflect the conventional computer science wisdom ("Brooks' Law") that software costs scale quadratically with the number of programmers involved (Brooks 1982). They are also consistent with empirical estimates of software costs, although some authors prefer a linear function (Dolado 2001). Finally, we assume that OS production has no inherent cost advantage over CS except to the extent that it allows members to share costs. This seems justified given scholars' rudimentary knowledge of this subject (Koch 2004, Asundi 2005).[14]

[13] The costs of total OS code $X^{\text{os}} = \sum x_i^{\text{os}}$ are $c(X^{\text{os}}) = \frac{1}{2}\phi X^{\text{os}2}$. As OS implies cost-sharing each firm bears only a fraction, related to its own code development x_i^{os}. Also, all individual costs must sum to total OS code costs, i.e. $c(X^{\text{os}}) = \sum c_i$. This yields $c_i = {}^{x_i^{\text{os}}}/_{X^{\text{os}}} \cdot {}^1/_2 \cdot \phi X^{\text{os}2} = {}^1/_2 \cdot \phi x_i^{\text{os}} X^{\text{os}}$. See also von Engelhardt (2010).

[14] Some commentators have argued that OS is inherently cheaper to develop than CS software (Raymond 1998). The current model could be readily extended to cover such scenarios if and when they are confirmed.

3.2. How Much OS and CS Software Does the Market Supply?

We now ask how firms choose to invest in OS and CS at stage one. For now, we treat the number of OS and CS firms as a free parameter without asking whether they represent equilibrium outcomes in real markets. (We will return to this question in Section 4.) It turns out that only two types of firms need to be considered: OS firms that use (and sometimes help to develop) open code; and CS firms for whom x_i is their own internally developed, proprietary code.[15]

Software increases the bundled good's quality and hence increases consumer demand. This gives firms an incentive to invest in software. The strength of this incentive depends on marginal sales (increased revenue per added code line) which in turn depend on each firm's ability to capture the social value of its improvements in stage two sales. When OS and CS firms compete, three types of Nash-Equilibria are possible depending on the model's parameters: OS-only production, simultaneous CS- and OS-code production, or CS-only code production.

3.2.1. A Pure-CS Industry

Consider first an industry in which no OS firms exist. How much CS is produced? In general, the answer depends on firms' strategic interactions, i.e. on how Firm A reacts to Firm B's decision to produce code. Solving for the amount of CS software x^{cs} that maximizes the profit function $\pi_i = p_i q_i - c_i - C$ leads to the following reaction function:

$$R_i^{cs} = \frac{(1+(n-1)\,\theta)\left(1-\theta\sum_{j\neq i} x_j^{cs}\right)}{\frac{1}{2}\phi h^2 - (1+(n-1)\,\theta)^2}, \tag{3}$$

[15] In principle, firms could also produce an OS-CS mixture at stage one. Except for very special cases, however, such scenarios produce Prisoners' Dilemma equilibria in which all firms seek to consume OS code but no firm supplies it (von Engelhardt 2010). We thus assume "viral" OS-licenses that prohibit such an OS-CS code mix. Most real world OS collaborations similarly outlaw mixed strategies by adopting GPL-type licenses.

This reaction function says that firms' software development decisions are strategic substitutes, i.e. Firm B's decision to increase output always encourages Firm A to decrease production and vice versa (von Engelhardt 2010). The size of this effect depends on θ, i.e. the extent to which firms compete on quality. Furthermore, an industry composed of n identical firms obeying (3) has the following Nash equilibrium:

$$x^{cs*} = \frac{(1+(n-1)\,\theta)}{\frac{1}{2}h^2\phi - 1 + (n-1)\,\theta} \tag{4}$$

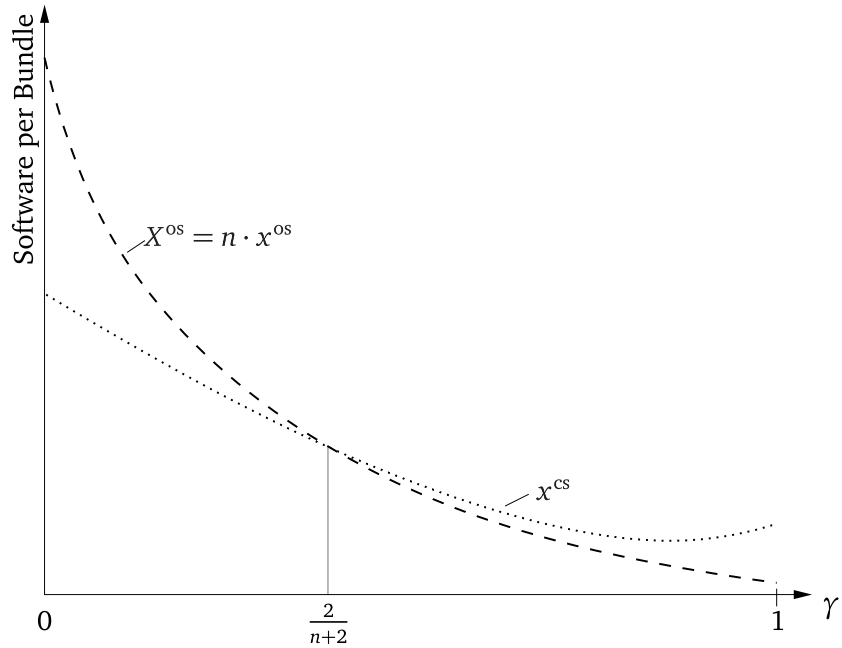
Here, code development is suppressed by intense quantity competition, i.e. the presence of h in the denominator. Conversely, quality competition—$(n-1)\theta$—increases equilibrium code-output by making the numerator larger and denominator smaller.

Both h and θ depend on the substitutability (γ) of the 'stage two'-products, while h also depends on the number of competitors (n). The net effect of an increase of n is straightforward: it decreases equilibrium code-output. However, changes in γ have a positive impact on both quantity competition and quality competition. Because quality and quantity competition exert opposing effects on CS output the net effect is more complicated. Figure 2 plots closed source (x^{cs}) production as a function of γ.

For low-to-moderate values of γ the amount of CS code produced is mainly determined by quantity competition (h), i.e. firms' ability to extract extra profits when quality increases. This ability is highest when products have no substitutes ($\gamma = 0$ yields $h = h^{min} = 2$) so that each firm can set monopoly prices unconstrained by competition. It steadily erodes as substitutability (γ)—and hence h—increases.

There is also a second effect determined by θ. For very large γ products are nearly identical so that even small quality differences can lead to large swings demand for or against a particular bundle. This makes quality competition extremely important. Specifically, CS firms find themselves in a kind of Arms Race or Prisoner's Dilemma in which each firm invests in quality to prevent every other firm from taking its business. This drives industry profits toward zero. This effect becomes so large for γ larger than ~ 0.9 that software

Figure 2: Software per Bundle in Case of Pure-OS and Pure-CS (stylized illustration)



production in a Pure-CS industry actually starts to rise again.[16]

Result 1. Software output in Pure-CS industries is suppressed by quantity competition but boosted by quality competition. Specifically, output fall as the number of competing firms (n) and/or product substitutability (γ) increases, hence h increases. However, at high γ quality competition (high θ) between firms actually increase output by creating an Arms Race dynamic in which each firm invests in quality to keep rivals from taking its business.

[16] Firms do not, of course, invest to the point where they would earn negative profits. For this reason, the effect is stronger in concentrated (small n) industries where firms possess sufficient market power to extract oligopoly profits. More precisely, CS starts rising at $\gamma = \left(n-5\sqrt{9+(n-2)n}\right)/2(n-2)$. This point is located at $\gamma = 0.8956$ for $n = 4$ and is higher for all other n. For large n software production only starts to rise near $n = 1$.

3.2.2. A Pure-OS Industry

Now consider the opposite case where no firms produce CS. Compared to CS industries, OS introduces two new effects. First, firms share cost. This means that the average per-firm development costs are lower for OS compared to CS firms. Second, OS firms share all software. This means that a firm's decision to invest in OS software not only makes its own bundles more attractive but also—contrary to the CS case—strengthens its competitors. Furthermore, the existence of shared software guarantees that no OS firm can offer better quality than any other OS firm. This suppression of quality competition implies that firms in Pure-OS industries always earn higher profits than firms in Pure-CS industries for a given number of incumbents. The same logic also implies that more firms will enter Pure-OS than Pure-CS industries. Ceteris paribus we therefore expect Pure-OS industries to have more incumbents when free entry is present.

As before, we start by asking how much software is produced. Solving once again for profit maximization, we find that each individual firm produces the following amount of OS:

$$R_i^{\text{os}} = \frac{1 - \left(\frac{1}{4}\phi h^2 - 1\right)\sum_{j\neq i} x_j^{\text{os}}}{\frac{1}{2}\phi h^2 - 1}. \tag{5}$$

Crucially—and unlike the Pure-CS case, see (3)—Firm B's decision to produce software no longer depends on quality competition (θ) from other firms. It does, however, depend on cost-sharing. This leads to several important differences from the CS case.

- *Strategic Complements vs. Substitutes:* For CS Firms, increased software output by Firm B always suppresses software for Firm A. By contrast, the result for OS firms is ambiguous. OS implies code- and cost-sharing. The net result is that firm decisions sometimes become strategic complements, i.e. that Firm B's decision to develop more software cause Firm A to increase its code output and vice versa. This occurs when the marginal cost of software production increases slowly (low ϕ) or competition is modest (low h). More formally, OS investments are strategic complements for $h^2\phi < 4$ (von Engelhardt 2010).

- *Quality-Cartel:* We have seen that quality competition among CS firms leads to Arms Races at high γ in which firms continue to invest in software until rising marginal costs wipe out any profits that could have been earned from increased demand. By contrast, OS firms do not compete on quality. Thus, they face no Arms Race, so that code output is suppressed to levels slightly below those that would be expected under a formal quality-cartel charged with setting output to whatever level maximizes total industry profit.[17]

As before, output decisions by n identical firms lead to a Nash equilibrium in which each firm produces the following amount of software:

$$x^{\mathrm{os}*} = \frac{1}{\frac{1}{4}\phi h^2(1+n) - n} \tag{6}$$

This implies that total industry-wide output ($X^{\mathrm{os}} = nx^{\mathrm{os}}$) is

$$X^{\mathrm{os}*} = \frac{n}{\frac{1}{4}\phi h^2(1+n) - n} \tag{7}$$

except in those cases where OS development would exceed the cutoff, i.e. deliver more code than society can use.[18] This leads to a third fundamental difference between the Pure-OS and Pure-CS models:

- *Quantity Competition and Cost-Sharing.* As in the Pure-CS case, the amount of software produced in Pure-OS industries depends negatively on quantity competition ($h = 2 + (n-1)\gamma$) and is greatest at low γ, see

[17] The difference stems from the fact that OS firms cannot recover the positive impact that their investment confers on the profits of the other OS firms. A formal quality-cartel would internalize this externality, see Appendix A.

[18] This formally occurs where $h^2 \cdot \phi \cdot {}^{(n+1)}/_{n} < 4$. In practice, this condition only occurs under relatively special parameters (von Engelhardt 2010) and we ignore it in what follows.

Figure 2. Now, however, there is a second effect. Because of shared development costs, Pure-OS industries are able to offer more software per bundle than Pure-CS industries as long as $\gamma < 2/(n+2)$.

- *Quantity vs. Quality Competition.* We have seen that quality competition gradually replaces quantity competition as the most important factor in determining CS output at high θ. Pure-OS industries, however, are able to suppress quality competition through code-sharing (quality-cartel). This explains why Pure-CS industries deliver more software than Pure-OS industries for $\theta > (4-h)/(2+h)$ or, equivalently, $\gamma > 2/(n+2)$, see Figure 2.

The balance between quantity competition and cost-sharing is superficially consistent with Llanes and de Elejalde (2009) and Henkel (2006)'s findings that OS business models are most profitable where quantity competition is low so that cost-sharing dominates. However, these earlier analyses are incomplete to the extent that they fail to consider the suppression of quality competition in Pure-OS industries. This latter effect systematically reduces the amount of OS output that would otherwise be expected from a simple balance of appropriability and cost-sharing.

Result 2. Cost-sharing allows Pure-OS industries to produce more software than Pure-CS industries so long as quantity competition is modest. However, production falls steeply as greater product substitutability (γ) leads to increased quantity competition between firms. Unlike CS firms, OS firms do not compete on quality (the 'quality-cartel effect'). For this reason Pure-OS industries offer less software per bundle than Pure-CS industries for $\theta > (4-h)/(2+h)$, i.e. $\gamma > 2/(n+2)$.

3.2.3. A Mixed OS/CS Industry

Finally, consider the case where both OS and CS firms exist. As before, CS firms still react to CS firms and OS firms still react to OS firms. Solving for this strategic interaction within each group yields equation (8), and (9) respectively[19]

$$x^{\mathrm{cs}} = \frac{(1+(n-1)\theta)\left(1-z^2\theta x^{\mathrm{os}}\right)}{\frac{1}{2}h^2\phi-(1+(n-1)\theta)(1+z\theta)} \tag{8}$$

$$x^{\mathrm{os}} = \frac{(1+r\theta)(1-\theta r x^{\mathrm{cs}})}{\frac{1}{4}\phi h^2(1+z)-z(1+r\theta)^2} \tag{9}$$

where z is the number of OS-firms and r is the number of CS-firms so that $n = r + z$. Now, however, CS firms also react to OS firms and vice versa. The overall Nash-equilibrium is thus a simultaneous solution of these two functions.

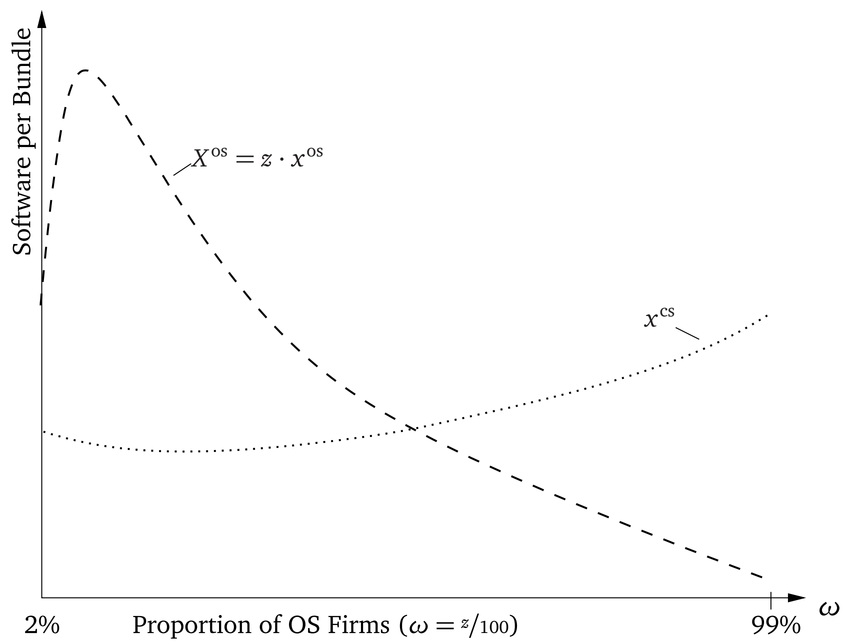
As before, CS firms react to increased software development by other firms as a strategic substitute. Since OS firms compete with CS firms on quality, they also see increases in CS production as strategic substitutes. This is represented by $-\theta r x^{\mathrm{cs}}$ in the numerator of (9). The situation regarding the strategic interaction among OS firms is different. The presence of CS firms prevents OS firms from cartelizing around a low level of quality. Instead, OS firms must compete on quality against these CS outsiders to the cartel. This makes the ability to share costs more valuable to OS firms. Ceteris paribus, this increases OS output and increases the strategic complements effect among OS firms. This explains why OS-investments are strategic complements for $h^2\phi < 4$ in Pure-OS industries but $h^2\phi < 4(1 + r\theta)^2$ for mixed OS/CS industries (see also von Engelhardt 2010).

Because of these interactions, the detailed behavior of a mixed OS/CS industry depends on the ratio of OS to CS firms. Figure 3 parameterizes this as the proportion of OS firms $\omega = z/n$ and shows how much software consumers receive per bundle in the typical example where $\gamma = 0.5$ and $n = 100$. Where OS firms are a small minority ($\omega \sim$ a few percent), they face strong quality competition from CS firms. This encourages them to use their cost-sharing advantage to produce large amounts of OS software. Increased opportunities

[19] As before, we exclude cases where OS would exceed the cutoff, i.e. deliver more code than society can use. Formally, we restrict our analysis to $\phi > 4 \cdot z/(z+1) \cdot (1+r\theta)^2/h^2$, see von Engelhardt (2010).

for cost sharing continue to dominate diminished quality competition until the proportion of OS firms reaches $\omega \sim 0.08$. Thereafter, however, the declining number of CS firms suppresses quality competition so that OS code production falls. For large ω the OS quality-cartel is so strong that OS firms produce very little code.

Figure 3: Software per Bundle in a Mixed Market ($n = 100, \gamma = 0.5$)



Because CS firms do not share costs, they cannot possibly match the maximum potential software output that OS firms can achieve. Furthermore, as already mentioned, CS firms react to OS development as a strategic substitute. As long as OS production is high, therefore, CS firms will specialize in selling low-quality bundles at a low price. The situation is reversed as OS productions declines. As a result, CS firms replace OS firms as the industry's high-quality, high-priced providers above $\omega \sim 50\,\%$.

Result 3. Quality competition from CS firms in mixed OS/CS industries mitigates the quality-cartel effect that suppresses production in Pure-OS industries. As a result, OS-software production for suitably chosen OS:CS ratios is dramatically higher than that found in otherwise comparable Pure-CS or Pure-OS industries. If the OS:CS ratio is low, OS firms are the industry's high-quality, high-priced providers. This situation is reversed when the industry hosts many OS firms and only a few CS firms.

3.3. Welfare Implications

We now know how much software a Pure-OS, Pure-CS, or Mixed-OS/CS industry produces. This allows us to calculate firm profits and consumer utility, which in turn enables us to analyze welfare. More specifically, producer surplus is given by total industry profits, and consumer surplus for differentiated oligopolies is given by $1/2\left[(1-\gamma)\sum q_i+\gamma(\sum q_i)^2\right]$ (see Hsu and Wang 2005). This yields the following general welfare function:

$$W=\frac{1}{2}\left[(1-\gamma)\sum q_i+\gamma\left(\sum q_i\right)^2\right]+\sum \pi_i \tag{10}$$

3.3.1. Pure-OS vs. Pure-CS

We begin by comparing welfare under a Pure-OS regime against a Pure-CS case. For convenience, we focus on the difference in welfare between a Pure-OS and a Pure-CS world. Prices and quantities of Pure-OS and Pure-CS are given by $q=p={(1+nx^{\text{os}*})}/{h}$ and $q=p={(1+x^{\text{cs}*})}/{h}$ respectively. The *difference* in welfare between a Pure-OS and a Pure-CS world is given by

$$\mathcal{W}_n=W_n^{\text{os}}-W_n^{\text{cs}}=\underbrace{\frac{n}{2}(1+h)\frac{\left(1+nx^{\text{os}*}\right)^2-\left(1+x^{\text{cs}*}\right)^2}{h^2}}_{\text{quality difference}}-\underbrace{\frac{n}{2}\phi\left(nx^{\text{os}*2}-x^{\text{cs}*2}\right)}_{\text{cost difference}}. \tag{11}$$

Note that the welfare difference consists of two components. The first reflects the *quality difference* and the second term represents the *cost difference* between a Pure-OS and a Pure-CS world. (For more details on the welfare functions see Appendix B)

We start by examining welfare at $\gamma = 2/(2+n)$ where, as we have already seen, Pure-OS and Pure-CS systems produce the same amounts of code (see Figure 2) so that the quality difference between firms' 'stage two'-products is zero. Here, the remaining cost difference term makes OS welfare superior. This is because OS firms can share code whereas each CS firm must create its own code base *de novo*. This wasteful duplication of effort is variously described as "business stealing" or "me-too products" in the literature (Henkel and von Hippel 2005).

OS's welfare-superiority diminishes for small values of γ. This is because Pure-OS industries produce much more code (see Figure 2) and therefore incur higher costs. However, the effect is never large enough to overcome the quality and cost-sharing advantages associated with OS. For this reason, Pure-OS industries remain preferable to Pure-CS industries in our model for all $\gamma < 2/(2+n)$. Significantly, this statement does not depend on ϕ and therefore holds regardless of detailed assumptions (e.g. Brooks' Law) about how quickly the marginal cost of software rises.

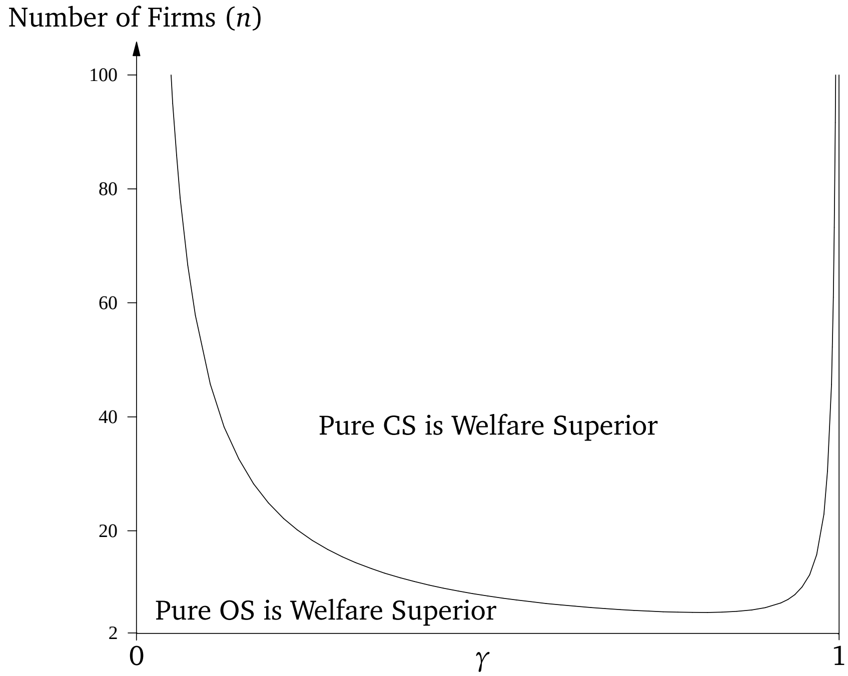
The situation is more ambiguous for γ larger than $2/(2+n)$. At first OS's welfare-superiority erodes with increasing γ because of the greater code production associated with Pure-CS industries (see Figure 2). For $\gamma > 2/(2+n)$ Pure-CS delivers higher quality. For moderate large γ this effect dominates the cost advantages of shared OS production so that CS also delivers superior welfare. On the other hand, we have seen that code output in Pure-CS industries increases sharply for very large values of γ. This can produce such large cost increases that a Pure-OS industry is once again welfare-superior.

Figure 4 summarizes these results for a representative numerical example in which we have set $\phi = 2$.[20] (We solve $\mathcal{W}_n = 0$ for n, with $\phi = 2$ and $\mathcal{W}_n$, x^{cs} and X^{os} given by (11), (4) and (7) respectively. The result is plotted in Figure 4.) Note in particular that OS is welfare-dominant in highly concentrated industries (low values of n), for limited substitutability (low values of γ), and situations where both n and γ are moderate. More concisely, OS is

[20]Our results do not change significantly for values different than $\phi = 2$, see Appendix C.

welfare-superior where $h = 2 + (n-1)\gamma$ is small. Furthermore, OS is also welfare-dominant for very high values of γ. Pure OS are thus superior both for low quantity competition and very high quality competition.

Figure 4: Welfare Superiority of OS- vs. CS-only for $n = 2 \ldots 100$ ($\phi = 2$)



Result 4. We find (a) that Pure-OS industries are welfare superior to Pure-CS industries for low values of quantity competition, (b) that Pure-CS industries are welfare-superior to Pure-OS industries for high values of quantity competition, and (c) that Pure-CS industries again become welfare-superior to Pure-OS industries when competing products are very close substitutes so that quality competition is intense.

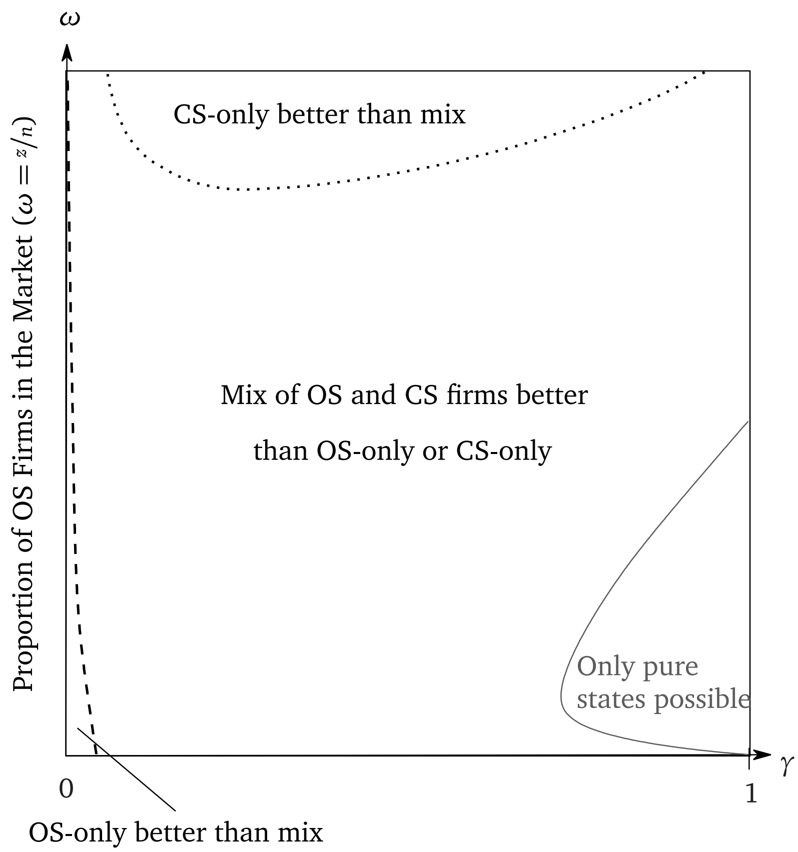
3.3.2. Mixed OS/CS-Industries

We now extend our welfare analysis to include arbitrary mixes of OS and CS firms where the proportion of firms creating OS is given by $\omega = \mathrm{z/n}$. We want to know whether Mixed OS/CS, Pure-CS, or Pure-OS industries generate more welfare. We therefore use (10) to calculate welfare for each pair (ω, γ), and compare this against our results for Pure-OS and Pure-CS industries. (For details on the welfare function see Appendix B). Figure 5 depicts our results for industries containing $n = 100$ firms, $\phi = 2$. (We plot welfare of Pure-CS, Pure-OS and Mixed OS/CS with $\phi = 2$ and $n = 100$. The topview of the resulting 3-D plots yields Figure 5.)

Figure 5 contains four distinct regions. The largest region consists of mixed states that are welfare-superior to the corresponding Pure-OS or Pure-CS state. Furthermore, readers can confirm by inspection that such welfare-superior mixed states exist for all values of γ. Thus, provided that ω (the proportion of OS-firms) can be suitably chosen mixed industries are the better choice. While we have depicted this situation for the special case of $n = 100$, this statement is in fact generally true for all large n, while for very concentrated industries (small n) the situation can differ (For some low value of γ welfare-superior mixed states might not exist as Pure-OS is superior for all ω. See Figure 13 and Figure 14 in Appendix D).

Sometimes, pure states offer higher welfare than mixed ones. This is reflected in the two smallest regions of the graph which are, in effect, much-shrunken versions of the Pure-OS and Pure-CS states depicted in Figure 4. First, consider the high ω region where OS firms greatly outnumber CS firms. Here, Pure-CS states are welfare-superior to mixed states for the same reasons that they dominate Pure-OS states. This region shrinks and eventually disappears as industry becomes more concentrated so that OS firms can appropriate value despite high γ (Figure 13 and Figure 14 in Appendix D). Second, Pure-OS is welfare-superior to mixed states in low ω/low γ cases where CS firms greatly outnumber OS firms and products have low substitutability. Here, OS firms can recover their investments even without intellectual property protection. As a result, cost-sharing dominates the welfare analysis so that Pure-OS states become superior. This region grows for concentrated (low

Figure 5: Welfare-Comparison of Pure vs. Mixed Cases ($n = 100, \phi = 2$)



n) markets until OS firms can recoup their investments even for moderate γ. The result is a drastically simplified graph in which the welfare-dominant regions are either Mixed-OS/CS or Pure-OS states (Figure 14 in Appendix D).

Finally, consider the lower right-hand corner of Figure 5. Here, only pure states are possible because (a) strong competition limits OS firms' ability to appropriate profits from improved products, and (b) the small number of

OS firms limits the savings otherwise available from shared development. This means that the profit-maximizing investment for OS-firms facing CS-competition is to produce no software at all.[21] As a result, the only meaningful choice is between Pure-CS and Pure-OS-states. Here, the Pure-CS state turns out to be welfare-superior (see Figure 4 for the case of $n = 100$).[22] This region also disappears for highly concentrated industries where competition is weak (Figure 14, Appendix D).

Welfare also differs within the mixed regions. For example, welfare in an $n = 100$ industry with $\gamma = 0.5$ reaches its maximum when OS-firms account for 20 % of all firms: While producer surplus reaches its maximum at $\omega = 99\,\%$, consumer surplus has an inverse U-shape with its peak at $\omega = 18\,\%$. The shape of consumer surplus is driven by the quality of the bundles available in the market and is maximized at a reasonable mix of a few high-quality, high-priced OS-bundles and many low-quality, low-priced CS-bundles. Thus, total welfare is maximized at a point where the OS firms select the much higher output levels when quality competition is supplied by a relatively high number of CS firms.

Result 5. Except for highly concentrated industries (low n), suitably-chosen mixed industries are always welfare-superior to Pure-OS or Pure-CS industries. These mixed states are usually welfare-superior because they feature enough OS firms to efficiently share costs and enough CS firms to provide quality competition. However, Pure-OS industries can still be welfare-superior to Mixed-OS/CS states where industry concentration is high.

4. Stable Outcomes in Case of Free Entry

We now know which situations (Pure-OS, Pure-CS, Mixed OS/CS) provide the most welfare. Here, we explore the extent to which markets actually deliver these outcomes. Except for Baake and Wichmann (2004) and Schmidtke (2006), previous contributions start from the assumption that industry size

[21] There is also a miniscule region where OS-firms develop code and CS-development is driven to zero. The area is located near the mixed cases border.

[22] Pure-OS is superior in a small number of cases where γ is very close to one.

is fixed (Casadesus-Masanell and Llanes 2009, Llanes and de Elejalde 2009, Henkel 2006, e.g.). Absent compelling empirical evidence, however, ignoring entry seems artificial. Furthermore, the proportion of OS-firms ($\omega = z/n$) and welfare implications of these models depend on initial assumptions about the number of firms. Leaving industry size a free parameter limits their predictive power.

We adopt a different approach. Specifically, we use our very general model to systematically identify and evaluate the specific combinations of incumbents (n) and the proportion of OS-firms (ω) that we expect to find in real markets. More specifically, we assume that industries will continue to evolve under entry until additional OS and CS can no longer earn a positive profit by entering the market.

Section 4.1 assumes that a Mixed-OS/CS industry already exists and asks which proportions of OS-firms are equilibria (i.e, resist further entry by OS and CS-firms). Significantly, we find that these proportions are systematically different from the target proportions that would be needed to maximize welfare. Section 4.2 then shows how industries can be locked into Pure-OS and Pure-CS states so that the transition to welfare-improving mixed industries never occurs.

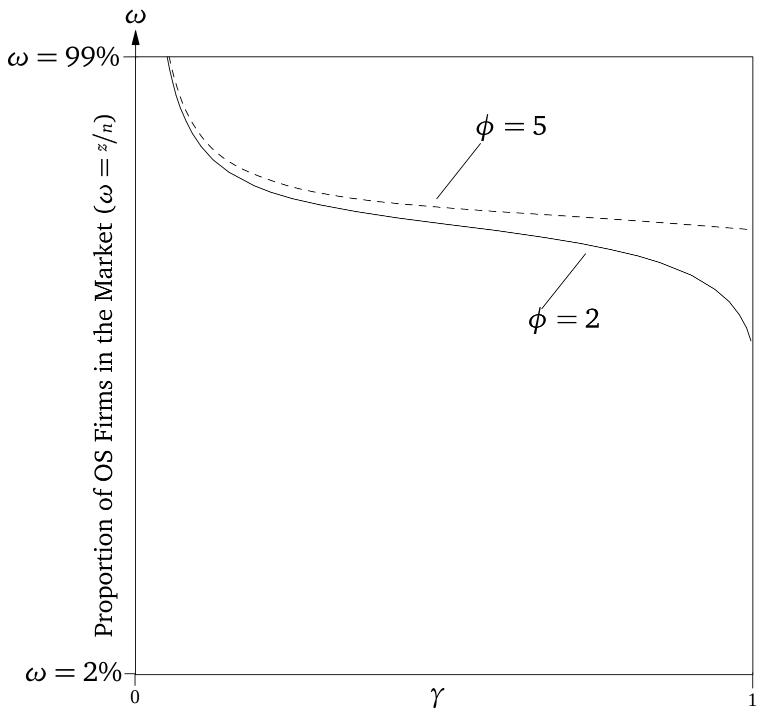
4.1. The OS:CS Ratio in Equilibrium

In order to be stable, an arbitrary mix of CS and OS firms must satisfy the following conditions: (a) incumbent OS firms a earn profit ≥ 0, (b) incumbent CS firms likewise earn a profit ≥ 0, and (c) would-be additional OS and CS firms cannot earn a profit ≥ 0 by entering the market. We start by analyzing the case in which the total number of firms is large. Because such markets are highly competitive, we expect each firm's (economic) profit to be identically zero:

$$\pi^{\text{os}} = p^{\text{os}} \cdot q^{\text{os}} - c_i^{\text{os}} - C = \pi^{\text{cs}} = p^{\text{cs}} \cdot q^{\text{cs}} - c_i^{\text{cs}} - C = 0.$$

Significantly, the necessary condition $p^{\text{os}} \cdot q^{\text{os}} - c^{\text{os}} = p^{\text{cs}} \cdot q^{\text{cs}} - c^{\text{cs}}$ depends solely on γ, ω, n and ϕ . Figure 6 displays the (γ, ω) pairs needed to ensure stability for a market containing $n = 100$ firms for $\phi = 2$ and $\phi = 5$. (We solve the necessary condition, with $n = 100$, for γ and plot the results of $\phi = 2$ and $\phi = 5$. Taking into account that x^{os}, $x^{\mathrm{cs}} \geq 0$ yields Figure 6.)

Figure 6: Entry-Resistant Proportion of OS Firms ($n = 100$)



Readers can confirm by inspection that the proportion of OS-firms (a) increases in ϕ, i.e. is higher where marginal costs increase steeply, and (b) decreases in γ, i.e. declines in industries where 'stage two'-products are close substitutes so that competition is high. The relatively large proportion of OS firms for most parameters occurs because any arbitrary number of OS firms is almost always more profitable than an equivalent number of CS firms in our model. This result is dramatically different from Llanes and de Elejalde (2009), who find that OS firms are only more profitable than CS firms when cost-sharing dominates appropriability. We attribute this difference to quality-cartel effects which systematically boosts OS profits in our model.

Despite this, CS firms still sell more bundles (i.e., have larger market share) than OS firms. Furthermore, as markets become more competitive (large γ), the average CS firm's market share grows while the average OS firm's market share shrinks. For $\gamma = 1$ the total number of bundles containing CS code greatly exceeds those containing OS code, see also von Engelhardt (2010). This result is strongly reminiscent of many industries (e.g. cell phones) where most consumers use products containing CS software even though OS firms greatly outnumber CS firms.

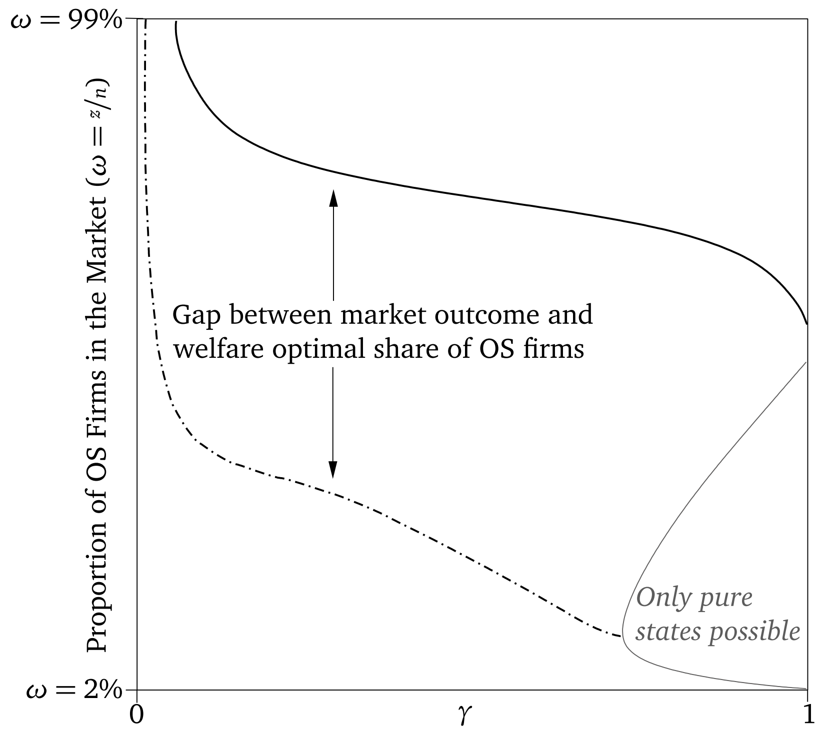
We have already seen that our model lets us calculate the welfare-optimal mix of OS and CS firms. Superimposing this plot on Figure 7 we see our stability condition (solid line) requires far more OS firms than the number needed to optimize welfare (dash-dotted line). (The market outcome is calculated as for Figure 6. To obtain the welfare optimal share of the $n = 100$, $\phi = 2$ case, we solve $\partial W/\partial \omega = 0$ for γ. Taking into account that $x^{\mathrm{os}}, x^{\mathrm{cs}} \geq 0$ and $\pi^{\mathrm{os}}, \pi^{\mathrm{cs}} = 0$ yields Figure 7.) This dramatic mismatch between the OS:CS ratio expected in equilibrium and the desired welfare-optimizing ratio is a central result of this article and poses an important challenge to policymakers.

The situation for concentrated (small n) industries is conceptually similar but more complicated. Since the number of CS and OS firms is always an integer,[23] the number of possible CS/OS ratios that can actually be realized in practice is small—for example, there are just four choices (0, $2/4$, $3/4$, 1) in a four-firm industry.[24] Thus, the stable OS:CS ratio predicted by our zero-profits analysis splits into two equilibria. For example in a four firm industry with $\gamma = 0.95$ our single predicted equilibrium at 0.69 can be realized by possible equilibria at 0.5 and also at 0.75. The 0.5 result improves welfare compared to our zero-profits calculation (0.69) while the 0.75 result makes it worse.

[23]Of course this condition holds even for large n. Nevertheless, it can usually be neglected unless n is small.

[24]Since OS only generates cost savings when there are two or more firms to share code, the $1/3$ solution is economically indistinguishable from the case of three CS firms.

Figure 7: Mixed Industry with $n = 100$ - Welfare Optimality vs. Market Outcome ($\phi = 2$)



Result 6. Mixed OS/CS industries are only stable against entry when incumbents earn zero (or, for concentrated industries, very limited) profit. Using this criterion, we find that there are typically far more OS than CS firms in mixed industries. In particular, the number of OS firms is very much greater than the number required to maximize welfare. CS firms, on the other hand, tend to have larger market shares than OS firms and this is especially true where firms' products are close substitutes (large γ).

4.2. Pure States and the Danger of Lock-In

We have seen that welfare optimality requires mixed OS/CS states. Suppose, though, that a particular industry starts off in a Pure OS- or CS-state. In this case, welfare improvements are bounded unless the industry can transition to a mixed state. Here, we explore the various circumstances under which industries can become locked in against such transitions.

4.2.1. Pure OS

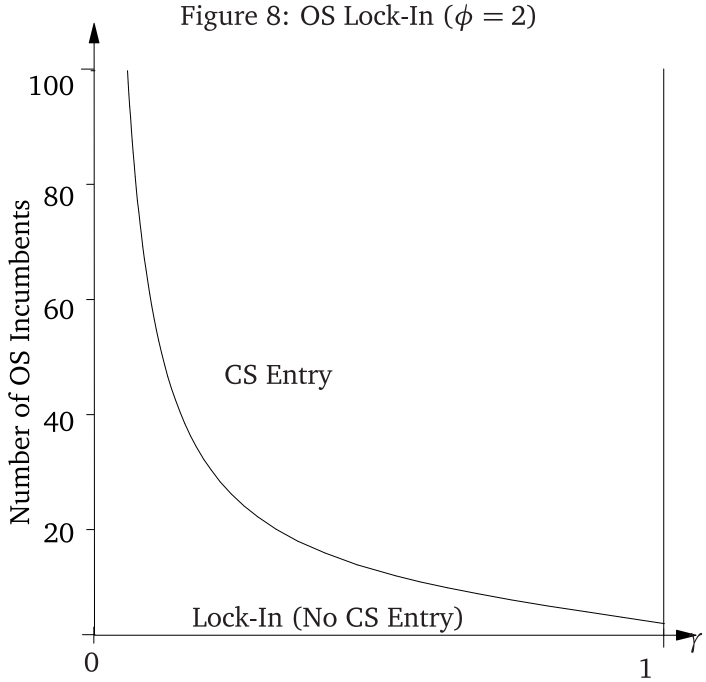
Suppose incumbents' profits in a Pure-OS industry are zero so that further OS entry is impossible. Our model allows us to calculate whether a CS entrant would earn a non-negative profit. Figure 8 displays this information for industries with different numbers of OS incumbents. (The zero profit condition for the z OS incumbents, $\pi^{\text{os}}_{n=z} = 0$, yields the corresponding stage two costs $C_{n=z} = p^{\text{os}}q^{\text{os}} - c^{\text{os}}_i$. CS-entry occurs only if the entrant's profits $\pi^{\text{cs}}_{n=z+1} = p^{\text{cs}}q^{\text{cs}} - c^{\text{cs}} - C_{n=z}$ are greater than zero. Solving $\pi^{\text{cs}}_{n=z+1} = 0$ for γ yields the boundary plotted in Figure 8.) This figure shows that industries can indeed be locked into a Pure-OS state that suppresses welfare. This only happens, however, where substitutability and/or the number of OS incumbents is small.

4.2.2. Pure CS

The case for OS firms' entry into a CS market is more subtle. Consider a Pure-CS industry in which incumbents' profits are zero. As before, we assume that this Pure-CS case is unstable if OS firms can earn a non-negative profit by entering. Since OS only confers economic benefits when firms are able to share costs, however, two or more firms must enter the market to reach a stable outcome. We therefore consider the case where two OS companies are willing to enter the market, and would earn a profit if both did so.[25] Then the first OS firm will not enter the market unless it knows that the second firm

[25] Our argument does conceptually not depend on this assumption and holds equally for "two or more" OS-entrants.

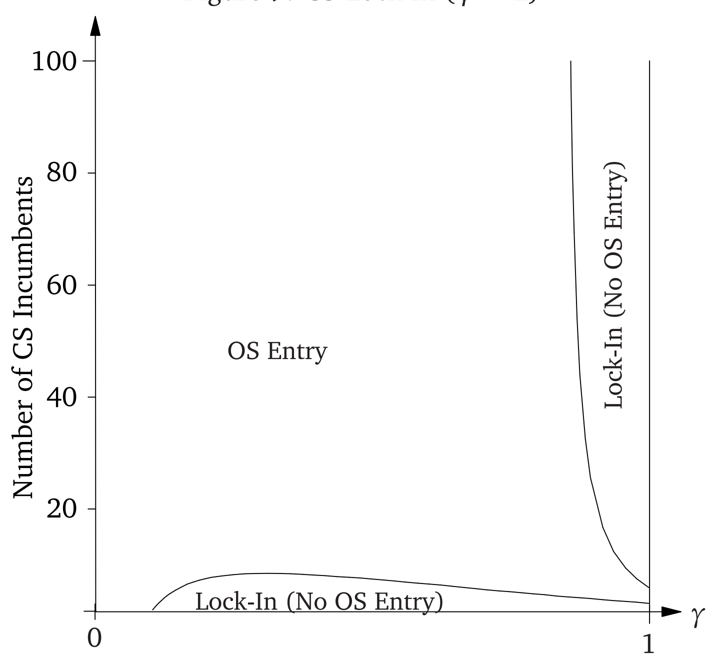
Figure 8: OS Lock-In ($\phi = 2$)



will also enter afterward. This means, by induction, that it will enter if and only if the second firm is assured of earning a profit. But this is exactly what the GPL license does. By making its own code GPL, the first entrant commits to the specific OS regime that allows the second entrant to earn a profit. We therefore conclude that there is no fundamental reason why Pure-CS states cannot transition to Mixed-OS/CS states so long as GPL-like commitment strategies exist. (For an alternative motivation see Appendix E)

The stability of the Pure-CS state depends on whether the two OS entrants can earn non-negative profits. Figure 9 depicts this situation. (The zero profit

Figure 9: CS Lock-In ($\phi = 2$)



condition for the r CS incumbents, $\pi^{\text{cs}}_{n=r} = 0$, yields the corresponding stage two costs $C_{n=r} = p^{\text{cs}}q^{\text{cs}} - c^{\text{cs}}$. OS-entry occurs only if the profits of each of the two entrants $\pi^{\text{os}}_{n=r+2} = p^{\text{os}}q^{\text{os}} - c^{\text{os}} - C_{n=r}$ are greater than zero. Solving

$\pi^{os}_{n=r+2} = 0$ for γ yields the boundary plotted in Figure 9.) While the Pure-CS state is unstable for most parameters, it is stable for (a) the concentrated, low n industries, most characteristic of Silicon Valley, and (b) industries whose products that are close substitutes.

4.2.3. Strategic CS Adoption by Incumbents

The preceding analyses make no assumptions about why a particular industry should find itself in a Pure-OS or Pure-CS state. Here we show that firms in some concentrated industries can deliberately stabilize their industry in a Pure-CS state. Readers should note that this strategy is only available to incumbents operating inside the ”No OS-Entry” regions of Figure 9. Outside these regions, OS firms can enter no matter what incumbents decide.

We explore this phenomenon by inserting a new stage 0 into our model. In this extension, incumbents decide whether to adopt OS or CS in stage 0a and e entrants decide whether or not to enter in stage 0b. The model then proceeds as before through stage one (incumbents and entrants calculate how much OS and CS software to develop) and stage two (firms develop a complementary product at cost C and decide how much to supply). As before, we define equilibria by the condition that new entrants would earn negative profit. In general, we expect incumbents to strategically adopt CS over OS whenever doing so will (a) block OS entrants, (b) block additional CS entrants, and (c) produce greater profits than incumbents would earn in a world where OS entry occurred.

To see how this extended model works, consider the example of a highly concentrated ($n = 4$) and differentiated ($\gamma = 0.3$) industry in the case where $C = 0.1055$. Absent entry, firms can earn profits of about 0.062 by adopting OS and nearly 0.023 by adopting CS. However, the OS profit is illusory since the savings from code-sharing would allow additional OS firms to enter the market until profits fall close to zero ($\pi = 0.000446$). By contrast, adopting CS is stable since in this case any new combination of CS and/or OS entrants would earn negative profits. We therefore expect our four incumbents to deliberately block entry by adopting CS. And this will be true even though entry following an OS decision would create an $n = 6$ Pure-OS industry that offers higher welfare. As usual, we expect these entry effects to be greatest for industries that are highly concentrated (small n) or feature low substitutability (low γ).

Result 7. Industries that start in a Pure-OS (-CS) state can be stable against the CS (OS) entry that would be needed to achieve a welfare-superior Mixed-OS/CS state. Incumbents may also adopt CS strategically in cases where OS would facilitate entry and erode oligopoly profits.

5. Government Intervention

So far, we have limited our analysis to asking how much software we expect OS and CS firms to supply in equilibrium. However, government can also intervene to change this private sector outcome. For example, we have stressed the role of OS as a de facto cartel that suppresses quality competition. In principle, government could redress this using antitrust (competitions) policy to render OS collaborations illegal. This, however, would eliminate OS even where we expect it to be welfare-superior. More fundamentally, it seems wasteful to discard OS's cost-sharing advantage unnecessarily. This section asks whether traditional policy levers based on taxation and preferred procurement policies can provide a better solution.

5.1. Tax Policy

This section examines how government can use lump-sum and progressive taxes and tax breaks to change industry output of OS and CS code.

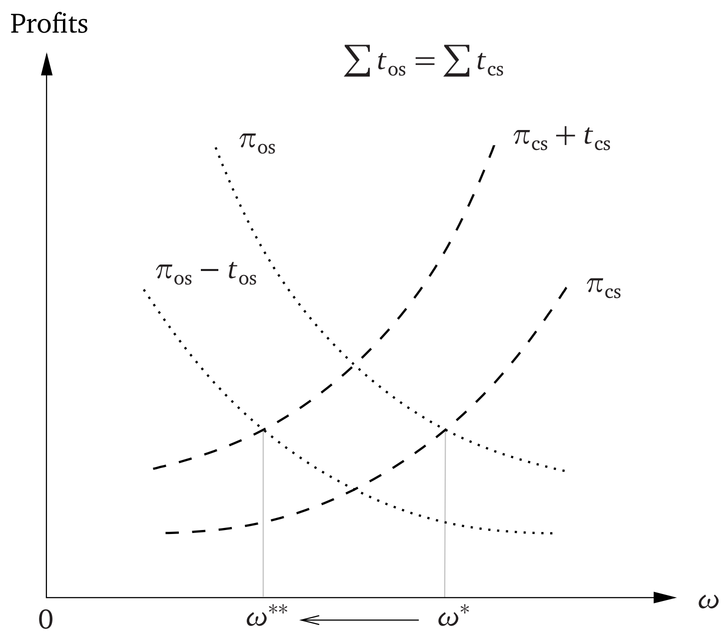
5.1.1. Discriminatory Lump-Sum Taxes

We have seen that market forces in Mixed-OS/CS industries invariably deliver far more OS firms than welfare-optimization requires. This suggests that the governments should view typical proposals to foster OS as an infant industry, say through special tax breaks, with suspicion. To the contrary: The most obvious way to improve welfare in our model is to promote CS firms instead. This is best done by taxing OS firms, giving tax breaks to CS firms, or both.

For concreteness, consider a scheme where government imposes a fixed, lump-sum tax on OS firms and uses the proceeds to give lump-sum tax breaks or subsidies to CS firms. Detailed calculation confirms the intuition that these interventions reduce the equilibrium ω in our model

Figure 10 presents a stylized illustration of this process. Here, suitable taxation adjusts the market outcome so that it coincides with the desired welfare-optimal proportion of OS firms. Furthermore, our hypothetical lump-sum taxes and tax breaks transfer profits from OS to CS firms while leaving total industry profits unchanged. This means that both the desired welfare-

Figure 10: Lump-Sum Tax on OS Firms and Tax-Breaks for CS Firms (Stylized Illustration)



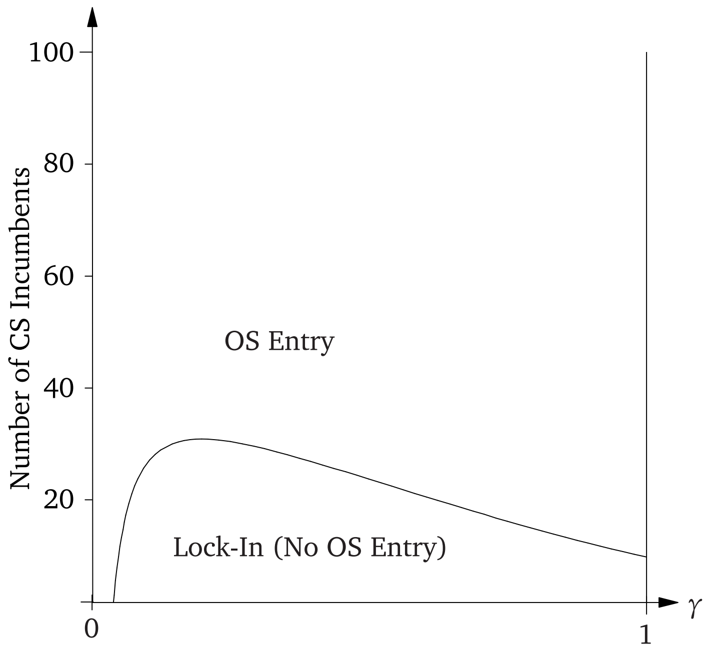
optimal OS:CS ratio and achievable welfare remain the same as before. This suggests that taxes and tax breaks are a potentially powerful tool. In practice, the main drawback is that government is not able to estimate the welfare optimal OS:CS ratio and/or proper taxation with any degree of precision. Ambitious transfer schemes could easily end up over-taxing OS firms and over-subsidizing their CS competitors. Additionally, per-firm profits after taxes and tax breaks could sometimes be negative, forcing government to make up the difference from general revenues.

5.1.2. Progressive Input-Tax

Lump sum taxes are not the only option. Government could also use progressive input-oriented taxation to change OS and CS-firms' effective marginal cost function to $(\phi + t)x$ where t denotes the tax. Detailed calculation shows, however, that this intervention does nothing to bring equilibrium OS:CS ratios in mixed industries closer to welfare optimality. The reason is that taxation simultaneously shifts both the OS:CS equilibrium ratio and the target welfare-optimal ratio in the same direction. This leaves the gap between the two just as wide as it was before.

On the other hand, changes in (effective) marginal costs do affect entry. This suggests that they could have a significant impact in helping Pure-CS or Pure-OS industries evolve into Mixed-OS/CS industries. Even here, however, value-added taxes provide an ambiguous intervention. To see this, compare a government tax policy that increases the slope of the (effective) marginal costs to $\phi + t = 5$ (Figure 11) compared to the $\phi + t = 2$ case depicted Figure 9. On the one hand, taxation has eliminated the right-hand (high γ) region where CS industries were formerly stable. This clearly facilitates transitions to welfare-improving mixed states. On the other hand, taxation also increases the second region near the bottom of the figure where CS is stable. This suggests that policymakers trying to destabilize a Pure-CS industry could accidentally reinforce it instead. This is particularly likely to happen for high γ/low n industries where the two regions are close to one another.

Figure 11: CS Lock-In in Case of $\phi + t = 5$



5.2. Government Provision of OS Software

Government can also intervene by paying contractors to create additional OS code over and above what the market supplies. We have already remarked that many countries have established different institutes, collaborations, grants, and partnerships to fund OS software development and adoption. The analysis is simplest for pure-CS markets where incumbents would otherwise be able to block entry by OS firms. Here, government-supplied OS reduces entry costs so that an OS sector can establish itself.

The case for government funding in mixed industries is more ambiguous because it takes place at two levels. The first effect takes place entirely within the private sector. Government-supported OS simultaneously (a) makes OS firms more profitable so that the equilibrium ω increases, and (b) reduces

OS firms' incentives to produce their own code (government OS crowds out private investment) so that the welfare-optimal ω falls. The net result is to make the already-large mismatch between the equilibrium and desired welfare-optimizing proportion of firms even worse. Indeed, sufficiently large government OS investments can drive CS firms out of the market entirely.

The second effect takes place across the entire economy. Suppose that government only cares about achieving the correct level of production without regard to how much the private sector contributes. Suppose further that government contractors have the same cost structure that the private OS firms do. Then the new, government-supplied code writers can be thought of as an additional OS firm that chooses output based on government fiat instead of our Cournot analysis. This allows policymakers to select any desired level of quality and still benefit from significant sharing with other private sector OS firms.[26] In principle, government can use this lever to increase welfare although it may be hard to estimate how much OS software to fund in practice.

5.3. Government Procurement Preferences

We have already noted that government is a major software purchaser. Many governments have tried to turn this spending into a policy instrument by establishing formal preferences (and even mandatory requirements) that systematically favor OS over CS products. This provides a powerful incentive for new OS entrants in cases where Pure-CS industries that would not otherwise evolve into welfare-improving mixed states.

In order to analyze the impact of preferences on mixed industries we add government demand to our model by assuming that government procures D bundles. We start with a neutral government as reference: Absent preferences, government buys one bundle from each firm so that each firm's demand is shifted by $d = D/n$. With government OS preferences, however, each OS firm sees an extra demand of $d = D/z$ while CS firms see $d = 0$. As result, the equilibrium proportion of OS firms increases. This widens the gap between the equilibrium ω and the ω needed to maximize welfare, just as with

[26] Llanes and de Elejalde (2009) similarly predict that government investment in OS increases the OS stock directly and also encourages more OS firms to enter the market.

government provision of OS code. However, unlike the case of government provided OS, this intervention reduces welfare. The new equilibrium ω has lower welfare than the equilibrium ω for a neutral government. Conversely, government procurement preferences for CS[27] yield the opposite results and increase welfare.

Result 8. Government can use various policy instruments to improve welfare by reducing high equilibrium proportions of OS firms. These include competitions policy (antitrust), taxation and tax break schemes, government-funded OS development, and government procurement preferences. Of these, tax policy provides the most natural instrument for achieving the target proportion needed to optimize welfare through private sector investment. By contrast, government-provided OS actually increases the gap between the desired and actual OS:CS ratio and depresses private OS investment still further. Despite this, government-funded OS still improves social welfare by boosting the total (private + government) supply of OS. Government procurement preferences for OS software have the worst outcome by not only increasing the gap between desired and equilibrium OS:CS ratio but also reducing total welfare.

6. Discussion

We have presented a general model that allows us to compare equilibrium OS:CS ratios against the target ratios that would be needed to maximize welfare. However, it remains possible that real markets may require more complex models. This section explores the extent to which such extended models would qualitatively change our results.

6.1. Basic Analysis

Our model OS output reflects a delicate balance between (a) lower per-firm costs through shared development, (b) reduced appropriability and hence

[27] In this case each CS firm has an extra demand of $d = {}^{D}/_{r}$ and CS firms have $d = 0$.

smaller investment incentives in case of strong quantity competition, and (c) a cartel effect that suppresses quality competition among OS firms. The first two effects are fairly straightforward and have previously been noted in, for example, Llanes and de Elejalde (2009), and Henkel (2006).

To the best of our knowledge, however, our third (”quality-cartel”) factor has never been noticed before. This is puzzling because Llanes and de Elejalde (2009)’s model is somewhat similar to ours. We conjecture that quality cartels do not appear in their model because they require firms to set quality and prices in a single, simultaneous decision. Introducing a single simultaneous decision similarly suppresses quality competition (and therefore quality-cartel effects) in our model.[28] We therefore expect a similar ”quality-cartel” effect to appear in an extended Llanes and de Eleljalde model—and, indeed, most generic models that require firms to make their quality and price decisions sequentially. Such models are also more realistic given the lags between most firms’ R&D (i.e. software) investment and pricing/output decisions.

6.2. Technology Assumptions

The quality of our bundles is completely determined by a single technology (software) which is (a) indivisible, (b) can be developed jointly, (c) can be shared, and (d) confers an identical quality “boost” on every product. In principle, all of these assumptions can be relaxed.

Our first assumption that all quality comes from a single indivisible technology (“software”) is more general than it looks since “software” can trivially be relabeled (a) to include other technologies (e.g. hardware) and (b) to exclude any technology (including GPL code) that is only usable by the author. Assuming that software is indivisible, however, does exclude situations where quality depends on multiple technologies each of which can be separately developed using OS or CS methods. One natural speculation is to ask what happens when firms can also invest in a second technology whose

[28] We analyzed a one-stage version of our model where firms choose the profit-maximizing (price, quality) pair. In the Pure-CS case quality competition is so weak that output continues to decline even at high γ. In the Pure-OS case, our results approximate the results for a formal OS quality-cartel described in Appendix A

benefits are primarily limited to their own ‘stage two’-product. Intuitively, we would expect this additional quality investment to drain resources from stage one R&D leading to fewer OS firms, less cost-sharing, and less development of shared software. This is more or less what happens when Llanes and de Eleljalde (2009) allow firms in their model to invest in a second technology focused narrowly on their products. The existence of a second, severable technology also facilitates strategic behaviors in which firms keep at least one technology closed as a barrier to entry (Schmidtke 2006).

Relaxing the second assumption that firms can develop code jointly is likely to be strongly model-dependent and should probably await convincing evidence that such failings actually exist. In the meantime, we note that Henkel (2006) has explored a model in which joint development is impractical at the level of individual OS modules so that each project is effectively controlled by one (and only one) company. Henkel argues that firms self-select toward developing whichever modules they value most and that this biases total OS investment upward. More generally, one can also imagine models in which OS joint development is possible but inefficient or imperfectly monitored. This could happen, for example, if participants adopted mixed strategies that encouraged them to strategically withhold effort from the collaboration in hopes that some other member would do the work (Johnson 2002)

The third assumption that firms can share OS might be relaxed if, for example, substantial ”tacit knowledge” was needed to use completed software. We therefore explored a variation of our model hat features a “spillover parameter” $\sigma \in]0,1]$, such that $\alpha_i = 1 + \sigma X^{os}$. We find that our OS results gradually converge to CS where spillovers are small.

Finally, completed software may not boost all firms’ products quality equally. Naively, we would expect the presence of some firms that gain relatively little from OS to produce free-rider effects. However, Henkel (2006) has shown that the existence of specialized interests within firms can actually increase total OS production. Relatedly, firms’ willingness to invest in OS could depend on the size of their respective stage two markets. It would be natural to investigate this by allowing different qualities of the ‘stage two’-products in our model. For now, it is probably safe to say that the answer will sensitively depend on how many separate technologies exist and the distribution of preferences among firms. Absent detailed empirical guidance, it will be hard to know which models to investigate.

6.3. Demand Side Assumptions

Our model assumes that consumers choose between products based on quantity supplied, substitutability, and a one-dimensional 'quality' parameter based on the amount of software produced. We recognize, however, that consumers may have idiosyncratic preferences for particular products. Naively, we expect strong consumer preferences to reduce the payoffs from quality improvements leading to lower code production. To the best of our knowledge, Llanes and de Elejalde (2009) are the only authors who have investigated firms' decisions to invest in quality using a Hotelling model that includes for idiosyncratic demand and allows for $n > 2$ firms. While their results are broadly similar to ours, Pure-OS industries are indeed much more common in their model. We conjecture that idiosyncratic consumer preferences reduce the importance of appropriability so that shared OS software production becomes more lucrative.

Similarly, the degree of substitutability (γ) is exogenous in our model. Over time, however, one might expect firms to design new products strategically so that γ becomes endogenous. This could be accomplished by, for example, linking our two appropriability variables n and γ. Alternatively, one might think that shared code would make OS products more similar (higher γ) to each other than to CS products. Llanes and de Elejalde (2009) explore this possibility by introducing different substitutability parameters for bundles that contain OS compared to bundles that contain CS. Not surprisingly, they find that increased substitutability leads to greater competition among OS firms which, in turn, makes CS firms larger and more profitable. We conjecture that endogenizing substitutability in our model would produce similar results.

Finally, we have limited our analysis to the case where products are substitutes. However, not all products compete and some are complements. As Schmidtke implicitly points out, OS provides a natural way for firms to encourage the production of complements that promise to increase their own product sales. Extending the current model to include this case would be reasonably straightforward. Doing so would probably mitigate the free rider but not the cartel effect. Furthermore, the number of such complementary products—and hence the importance of Schmidtke's observation—remains unclear.

6.4. 'Spooky' OS Incentives

We started this paper by remarking that commercial incentives have increasingly crowded out other incentives. However, volunteer labor remains important for many OS collaborations and dominate some. In principle, some of this voluntarism may reflect the desire for future wages and could in principle be endogenized in our model as a kind of prize. In general, however, many motives (reputation, altruism, fugue-state) are likely to remain forever outside predictive modeling. Following Einstein's reference to "spooky action at a distance" (Pais 1983) we refer to such non-commercial incentives as 'spooky OS' in what follows.

The effects of spooky OS are similar to those already considered for government provided OS. Thus, we expect increased spooky OS to (a) increase the total supply of OS, (b) increase the total number of OS firms, and (c) reduce the total amount of OS produced for commercial reasons. The main difference is that spooky OS potentially increases welfare faster than government OS. The reason is that programmers who gain psychic benefits from voluntarily supplying spooky OS have already been compensated because of their intrinsic motives (Lakhani and Wolf 2005, Hertel et al. 2003, Ghosh et al. 2002b). By definition, such hobby activities have no opportunity costs and thus increase welfare even more (von Engelhardt and Pasche 2004).

7. Conclusions and Outlook

Today's OS is increasingly dominated by business strategies in which firms make proprietary products whose quality depends on a shared OS code base.

We have presented a generic Cournot model based on simple, realistic assumptions about the costs of developing OS and CS code. We find that Pure-OS industries are welfare-superior to Pure-CS states so long as quantity competition is modest so that appropriability is high. Otherwise, Pure-CS industries are welfare-superior except for some industries where cost functions rise so steeply that CS firms are forced deep into diminishing returns in the case of nearly-identical products and strong quality competition. Finally, mixed industries containing a suitably-selected proportion of OS firms are superior to both Pure-OS and Pure-CS industries. This is because the presence of CS firms introduces quality competition which induces OS firms to produce more code than they otherwise would. Ironically, then, OS is only able to realize the full benefits of cost-sharing when CS firms are present.

Unfortunately, we find no evidence that the OS:CS ratios needed to maximize welfare are ever realized in practice. Instead, equilibrium mixed states consistently produce too many OS firms. Additionally, many pure markets are stable and cannot transition to the mixed markets needed to improve welfare. In some cases, incumbents can also deliberately block entry by choosing CS.

Policymakers trying to improve welfare can cope with these problems in several ways. Probably the most elegant (and also revenue-neutral) proposal is to tax OS firms and use the proceeds to subsidize their CS rivals. A more politically-correct—and ambitious—alternative would be for government to fund OS production directly. This would increase welfare over the market outcome although it also widens the gap between the equilibrium and welfare optimal proportion of OS firms. By contrast, procurement preferences that concentrate government spending on OS bundles should be rejected. Such schemes invariable decrease welfare and make the mismatch between the equilibrium and welfare-optimal proportion of OS firms even worse.

Appendix A OS versus a ‘Real’ OS Quality-Cartel

Let us compare the outcome of individual investment decisions by OS firms (the normal case analyzed in our model) against a hypothetical formal cartel in which firms agree to coordinate their activities to maximize joint profits. We find that individual decisions produce less output than a ‘real’ OS cartel would supply. Because of this underprovision/public good problem, the suppression of quality competition among OS firms only leads to a ‘second best’ cartel compared to the outcome under a ‘real’ quality-cartel.

A.1 OS-Firms, No Formal Cartel

In case of OS with competition, each OS firm $i = 1 \dots n$ maximizes its profits. Firm i’s reactionfunction is given by (5). By symmetry this same reaction function governs all $i = 1 \dots n$ firms. This determines the equilibrium value of $X^{\mathrm{os}} = \sum_i x_i^{\mathrm{os}} = n \cdot x^{\mathrm{os}}$ as given by (7):

$$X^{\mathrm{os}*} = \frac{n}{\frac{1}{4}\phi h^2 (1+n) - n}$$

A.2 A Formal OS Quality-Cartel

Consider a ‘real’ cartel in which firms develop code $X^{\mathrm{os}} = \sum_i x_i^{\mathrm{os}}$ so as to maximize joint profits $\Pi = \sum_i \pi_i^{\mathrm{os}}$. Then profit maximization requires:

$$\max_{X^{\mathrm{os}}} \Pi = \sum_i \pi_i^{\mathrm{os}} = n \frac{(1+X^{\mathrm{os}})^2}{h^2} - \frac{1}{2}\phi X^{\mathrm{os}2}$$

The zero of the first derivative then yields the cartel-output given by

$$X^{\mathrm{os,\ cartel}} = \frac{n}{\frac{1}{2}\phi\, h^2 - n}\ \forall \phi > \frac{2n}{h^2}$$

A.3 Comparing Outcomes

For all $\phi > \frac{2n}{h^2}$ (the second order condition of the cartel), the cartel produces more code than individual firms do:[29] $X^{\text{os, cartel}} > X^{\text{os}*}$. Furthermore firms also earn higher profits, i.e. $\pi_i^{\text{os, cartel}} > \pi_i^{\text{os}*}$. Finally, in terms of welfare we have $W^{\text{os, cartel}} < W^{\text{os}*}$.

Appendix B Welfare

Our analysis is based on Hsu and Wang (2005) who provide a general welfare analysis for differentiated oligopolies. Applied to our model, this yields the *consumer surplus A* as follows:

$$A = \frac{1}{2}\left((1-\gamma)\left(zq_{i\in Z}^2 + rq_{i\in R}^2\right) + \gamma\left(zq_{i\in Z} + rq_{i\in R}\right)^2\right)$$

with

$$q_{i\in Z} = \frac{\left(1 + zx^{\text{os}*} - r\theta\left(x^{\text{cs}*} - zx^{\text{os}*}\right)\right)}{h},$$

$$q_{i\in R} = \frac{\left(1 + x^{\text{cs}*} - z\theta\left(zx^{\text{os}*} - x^{\text{cs}*}\right)\right)}{h}.$$

The *producer surplus B* is given by

$$B = z \cdot \pi_{i\in Z} + r \cdot \pi_{i\in R},$$

with profits

$$\pi_{i\in Z} = \frac{\left(1 + zx^{\text{os}*} - r\theta\left(x^{\text{cs}*} - zx^{\text{os}*}\right)\right)^2}{h^2} - \frac{1}{2}\phi z x^{\text{os}*2},$$

$$\pi_{i\in R} = \frac{\left(1 + x^{\text{cs}*} - z\theta\left(zx^{\text{os}*} - x^{\text{cs}*}\right)\right)^2}{h^2} - \frac{1}{2}\phi x^{\text{cs}*2}.$$

[29] If the second-order condition for cartels is not satisfied, i.e. $\phi < \frac{2n}{h^2}$, the cartel produces software up to the cut-off.

B.1 Pure Cases

In case of Pure-OS ($n = z$) we obtain

$$W_n^{\text{os}} = A_{n=z} + B_{n=z} = (1+h)\,\frac{n}{2}\frac{\left(1+nx^{\text{os}*}\right)^2}{h^2} - \frac{1}{2}\phi\,\left(nx^{\text{os}*}\right)^2.$$

In case of Pure-CS ($n = r$) we obtain

$$W_n^{\text{cs}} = A_{n=r} + B_{n=r} = (1+h)\,\frac{n}{2}\frac{\left(1+x^{\text{cs}*}\right)^2}{h^2} - n\frac{1}{2}\phi\,x^{\text{cs}*2}.$$

We can now calculate the *difference in welfare* ($\mathcal{W}$) for a given n:

$$\mathcal{W}_n = W_n^{\text{os}} - W_n^{\text{cs}} = \frac{n}{2}(1+h)\,\frac{\left(1+x^{\text{os}*}n\right)^2 - \left(1+x^{\text{cs}*}\right)^2}{h^2} - \frac{n}{2}\phi\,\left(nx^{\text{os}*2} - x^{\text{cs}*2}\right).$$

with $nx^{\text{os}*} = X^{\text{os}*}$ is given by (7), and $x^{\text{cs}*}$ is given by (4).

B.2 Mixed Cases

The simultaneous solution of (8) and (9) is given by

$$x^{\text{cs}*} = \frac{(1+(n-1)\,\theta)\left(z^2\theta\,(1+\theta r) - \chi\right)\beta}{(1+\theta r)\,\theta^2 r\,(1+(n-1)\,\theta)\,z^2 - \psi\chi},$$
$$x^{\text{os}*} = \frac{(1+\theta r)\,(\theta r\,(1+(n-1)\,\theta) - \psi)\,\beta}{(1+\theta r)\,\theta^2 r\,(1+(n-1)\,\theta)\,z^2 - \psi\chi},$$

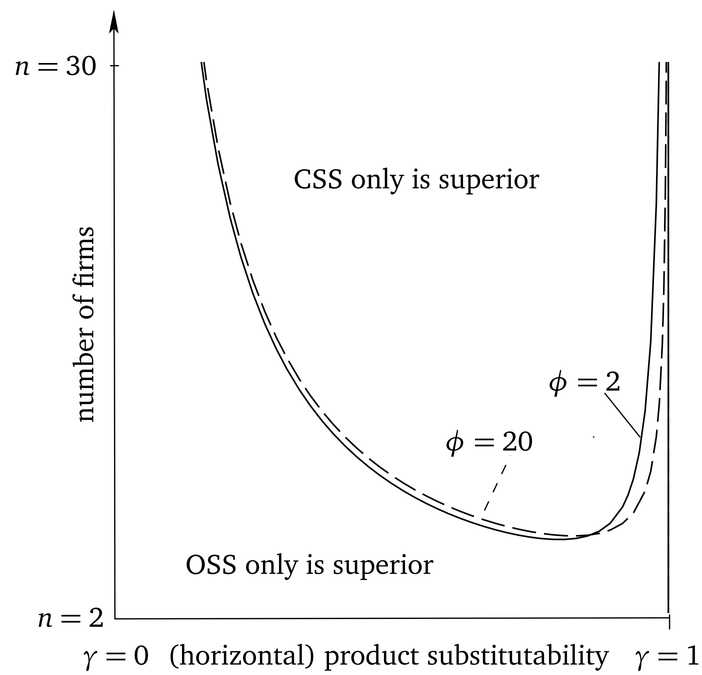
where ψ and χ are the denominators of (8) and (9): $\psi = \frac{1}{2}h^2\phi - (1+(n-1)\,\theta)(1+z\theta)$ and $\chi = \frac{1}{4}\phi h^2\,(1+z) - z\,(1+r\theta)^2$. Inserting these in our quantity and profits functions (see above) lets us numerically calculate total welfare according to the expression

$$W = \frac{1}{2}\left((1-\gamma)\left(zq_{i\in Z}^2 + rq_{i\in R}^2\right) + \gamma\left(zq_{i\in Z} + rq_{i\in R}\right)^2\right) + z\cdot\pi_{i\in Z} + r\cdot\pi_{i\in R}.$$

Appendix C Different ϕ and Welfare Comparison of Pure Cases

Here we show that adopting values of ϕ different from 2 does not substantially change our welfare comparison of Pure-OS and Pure-CS industries. Figure 12 shows how our $\phi = 2$ results (solid line) and $\phi = 20$ results (dashed line), change for different degrees of industry concentration ranging from $n = 2$ to $n = 30$.

Figure 12: Welfare Superiority of $\phi = 2$ vs. $\phi = 20$



As the reader can see, the situation does not change much when ϕ increases by a factor of ten. While the region where OS is superior expands slightly for low values of γ and contracts slightly for γ near 1 the qualitative results change very little. (Notice that we have exaggerated the differences by drawing the figure so that the affected regions are magnified compared to the figure in our main text.) The reason for this similarity is that changes in ϕ affect the cost function for Pure-OS and Pure-CS industries identically. Furthermore, the values of γ where (a) $X^{os} = x^{cs}$ and (b) where x^{cs} has its minimum (and begins to rise again) are given by (a) $\gamma = 2/(n+2)$ and (b) $\gamma = \left(n-5+\sqrt{9+n(n-2)}\right)/2(n-2)$ and hence independent of ϕ.

Appendix D Welfare of Pure vs. Mixed Cases in Concentrated Industries

Figures 13 and 14 show how our welfare analysis of Pure vs. Mixed cases changes for concentrated (small n) industries.

Figure 13: Welfare-Comparison of Pure vs. Mixed Cases, $n = 30$

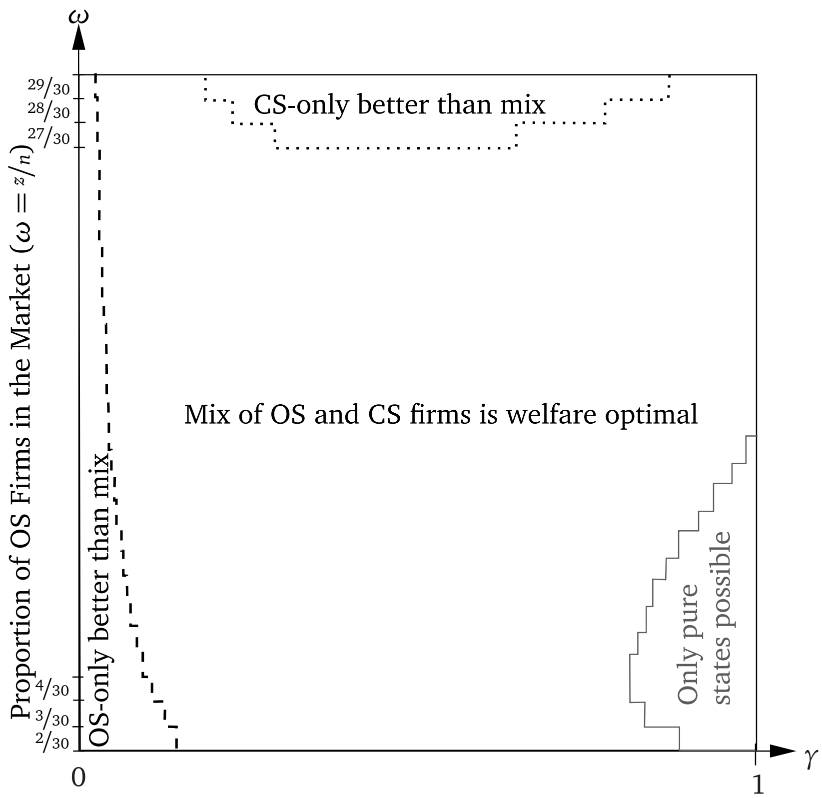
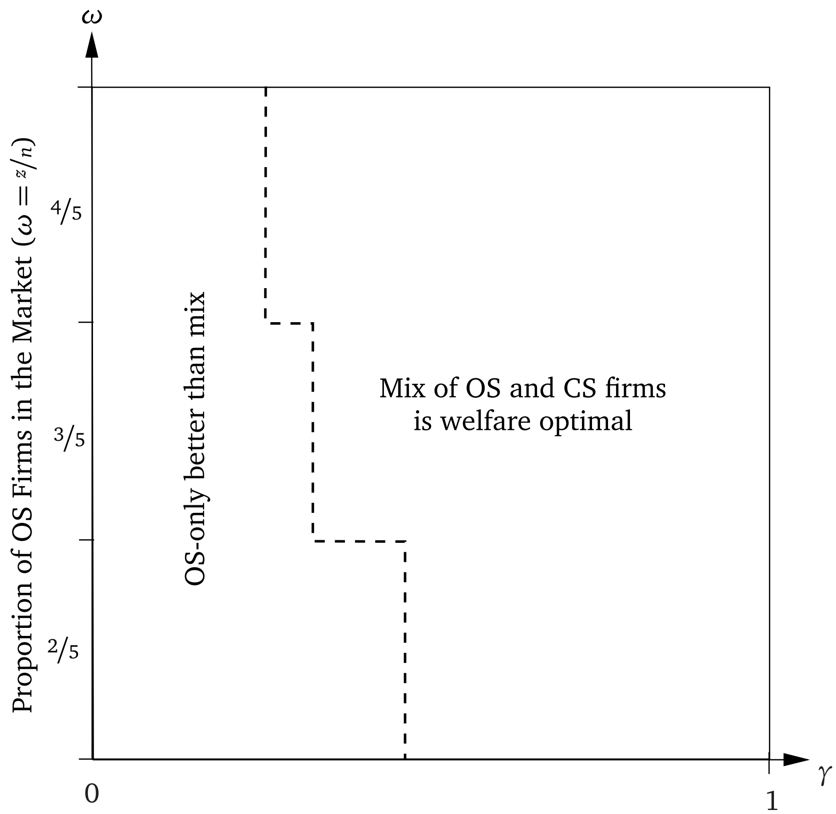


Figure 14: Welfare-Comparison of Pure vs. Mixed Cases, $n = 5$



Appendix E Alternative Motivation of OS-Entry into a Pure-CS Industry

Here we discuss simultaneous decision of potential OS entrants. Assume that an industry contains $n > 2$ incumbents, all of whom follow CS business models. Assume further that each incumbent's profits are zero so that further CS entry would produce negative profits. Finally, assume that there are $e \geq 2$ potential OS entrants. Each OS firm faces the following YES/NO choice on whether to enter.

For simplicity, we consider the case where $e = 2$ although our argument also holds for $e > 2$ entrants. Suppose that the $e = 2$ OS firms will earn a positive profit if they enter. Then the strategic problem is to ensure that both firms enter. (If only one OS firms enters, code-sharing cannot occur and the entrant will become a de facto CS firm earning negative profits.) This problem can be analyzed in terms of the following game where the payoffs have been normalized to 1, 0, and -1:

	YES	NO
YES	1,1	-1,0
NO	0,-1	0,0

This coordination game has two Nash-equilibria (YES, YES) and (NO, NO). Furthermore, this is a common interest game in which both two potential entrants would like to occupy the same (YES, YES) equilibrium. If players can signal which strategy they wish to play, we can assume that they will both arrive the common interest equilibrium. In our OS problem, this can readily be done if each OS entrant announces that its code will be subject to GPL.

References

Asundi, J.: 2005, The need for effort estimation models for open source software projects, *Proceedings of the Fifth Workshop on Open Source Software Engineering*, ACM, New York, NY, USA, pp. 1–3.

Baake, P. and Wichmann, T.: 2004, Open source software, competition and potential entry, *Berlecon Research Papers 5*, Berlecon Research, Berlin.

Brooks, F. P.: 1982, *The Mythical Man-Month: Essays On Software Engineering*, Addison-Wesley.

Casadesus-Masanell, R. and Llanes, G.: 2009, Mixed source, *NET Working Papers 09-06*, NET Institute.

CSIS: 2008, Government open source policies. Center for Strategic and International Studies. Available at `http://csis.org/`.

d'Aspremont, C. and Jacquemin, A.: 1988, Cooperative and noncooperative R&D in duopoly with spillovers, *American Economic Review* **78**(5), 1133–37.

Deshpande, A. and Riehle, D.: 2008, The total growth of open source, *in* B. Russo, E. Damiani, S. Hissam, B. Lundell and G. Succi (eds), *Open Source Development, Communities and Quality*, Vol. 275 of *IFIP International Federation for Information Processing*, Springer, pp. 197 – 209.

Dixit, A.: 1979, A model of duopoly suggesting a theory of entry barriers, *Bell Journal of Economics* **10**(1), 20–32.

Dolado, J. J.: 2001, On the problem of the software cost function., *Information & Software Technology* **43**(1), 61–72.

von Engelhardt, S.: 2010, Quality competition or quality cooperation? License-type and the strategic nature of open source vs. closed source business models, *Jena Economic Research Papers 2010-034*, Friedrich-Schiller-University Jena and Max-Planck-Institute of Economics.

von Engelhardt, S. and Pasche, M.: 2004, Volkswirtschaftliche Aspekte der Open-Source-Softwareentwicklung, *Jenaer Schriften zur Wirtschaftswissenschaft*.

Ghosh, R. A., Glott, R., Krieger, B. and Robles, G.: 2002a, FLOSS final report, part 3. Basics of open source software markets and business models, *Technical report*, International Institute of Infonomics, University of Maastricht.

Ghosh, R. A., Glott, R., Krieger, B. and Robles, G.: 2002b, FLOSS final report, part 4. Survey of developers, *Technical report*, International Institute of Infonomics, University of Maastricht.

H ckner, J.: 2000, A note on price and quantity competition in differentiated oligopolies, *Journal of Economic Theory* **93**(2), 233–239.

Henkel, J.: 2006, Selective revealing in open innovation processes: the case of embedded linux, *Research Policy* **35**(7), 953 – 969.

Henkel, J. and von Hippel, E.: 2005, Welfare implications of user innovation, *The Journal of Technology Transfer* **30**(1-2), 73–87.

Hertel, G., Niedner, S. and Herrmann, S.: 2003, Motivation of software developers in open source projects: an internet-based survey of contributors to the linux kernel, *Research Policy* **32**(7), 1159 – 1177.

Hsu, J. and Wang, X.: 2005, On welfare under cournot and bertrand competition in differentiated oligopolies, *Review of Industrial Organization* **27**(2), 185–191.

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

Koch, S.: 2004, Profiling an open source project ecology and its programmers, *Electronic Markets* **14**(2), 77–88.

Lakhani, K. R. and Wolf, R. G.: 2005, Why hackers do what they do: understanding motivation and effort in free/open source software projects, *in*

J. Feller, B. Fitzgerald, S. A. Hissam and K. R. Lakhani (eds), *Perspectives on Free and Open Source Software*, MIT Press, pp. 3 –22.

Lerner, J. and Tirole, J.: 2002, Some simple economics of open source, *Journal of Industrial Economics* **50**(2), 197–234.

Llanes, G. and de Elejalde, R.: 2009, Industry equilibrium with open source and proprietary firms, *Harvard Business School Working Papers 09-149*, Harvard Business School.

Maurer, S. M. and Scotchmer, S.: 2004, Procuring knowledge, *in* G. D. Libecap and M. C. Thursby (eds), *Intellectual Property and Entrepreneurship*, Advances in the Study of Entrepreneurship, Innovation, and Economic Growth, Emerald Group Publishing, pp. 1–31.

Pais, A.: 1983, *Subtle is the Lord: The Science and Life of Albert Einstein*, Oxford University Press.

Raymond, E. S.: 1998, The cathedral and the bazaar, *First Monday* **3**(3). `http://firstmonday.org`.

Schmidt, K. M. and Schnitzer, M.: 2003, Public subsidies for open source? Some economic policy issues of the software market, *Harvard Journal of Law & Technology* **16**(2).

Schmidtke, R.: 2006, Private provision of a complementary public good, *CESifo Working Paper Series CESifo Working Paper No.*, CESifo Group Munich.

Scotchmer, S.: 2004, *Innovation and Incentives*, The MIT Press.

Singh, N. and Vives, X.: 1984, Price and quantity competition in a differentiated duopoly, *RAND Journal of Economics* **15**(4), 546–554.

Verani, S.: 2006, Open source development in a differentiated duopoly, *Economics Discussion / Working Papers 06-05*, The University of Western Australia, Department of Economics.