Competition, Innovation and Intellectual Property Rights in Software Markets [(*)]

Michiel BIJLSMA, Paul W.J. DE BIJL & Viktória KOCSIS
CPB Netherlands Bureau for Economic Policy Analysis, the Netherlands

Abstract: This paper analyzes when it may be desirable for the government to stimulate open source software as a response to market failures in software markets. Our most important finding is that directly stimulating open source software, e.g. by acting as a lead customer, can improve dynamic efficiency if (i) there is a serious customer lock-in problem, while (ii) to develop the software, there is no need to purchase specific, complementary inputs at a substantial cost, and (iii) follow-on innovations are socially valuable but there are impediments to contractual agreements between developers that aim at realizing such innovations.

Key words: software markets, intellectual property rights, open source software, public policy.

In Europe, software falls under a copyright regime [1] [2]. Copyrights, and more generally intellectual property rights (IPRs), grant developers a temporary, exclusive right to reproduce and distribute the underlying code, as well as to develop derived works[3]. For software, one can distinguish proprietary and open source licensing. In the case of proprietary software (PS), the underlying source code is typically not disclosed so that tinkering

(*) This paper is based on BIJLSMA et al. (2009), a report commissioned by the Ministry of Economic Affairs (the Netherlands). We gratefully acknowledge extensive comments by colleagues at CPB, participants at the annual conference of NOiV (Utrecht, 5 March 2009), and two anonymous referees for useful comments. The usual disclaimer applies.

1 In the US, software is protected by patents, which pertain to ideas and come into existence after an application filed to the patent office has been granted. Copyrights aim at protecting the embodiment of ideas and apply automatically. Although we focus on copyrights, the thrust of our reasoning also applies to patents.

2 Arguably, copyrights are not perfect, since they protect software as a work of art or literary work, and therefore might not offer sufficient protection against reverse engineering. Instead, software developers may rely on trade secrets (i.e., not releasing the source code except under strict conditions, combined with penalties in case of breach of agreement or confidentiality), which are not recognized as IPRs.

3 See GALLINI & SCOTCHMER (2002), GUIBAULT & VAN DAALEN (2005), McGOWAN (2005) and STIGLITZ (forthcoming).

with the software by others cannot take place. Consumers or firms can only use a particular program with permission of the license owner. To obtain consent, users have to accept the license conditions and, in many cases, have to pay for the use of the software. Using or distributing the software without the permission of copyright owner constitutes a violation of copyright law. Open source software (OSS) "[...] has freely available source code enabling the licensee to inspect, use, improve, expand and distribute the source code" [4], under the conditions specified by the original license. OSS licenses typically use a watered down version of copyrights, in the sense that the licensor claims less rights than he can potentially claim. Just as in the case of PS, not complying with the license conditions violates copyright law.

From an economic point of view, the traditional idea behind IPRs is to protect the incentives for innovation by ruling out unauthorized copying and imitation. Therefore, one might think that OSS causes innovation processes to stall. Interestingly, this is not necessarily the case. In a recent empirical paper, LERNER (2009) reports a negative relationship between patent protection and innovation. As STIGLITZ (forthcoming) argues "[...] many of the most important intellectual advances are not covered at all by the patent system" (p. 105). BESSEN & MASKIN (2004) observe, related to the growth of the Internet, that "[...] the economic model underlying this traditional argument is surprisingly limited. […] Indeed, innovation is often sequential, where each creator improves on the work of the previous iteration" (p. 2). The basic idea is that when the innovation process is interactive and sequential, OS development may stimulate this innovation, which it ultimately helps to create value for society. BOLDRIN & LEVINE (2009) challenge the conventional wisdom with a theory of diminishing returns in knowledge creation; if knowledge creation is cumulative, patents reduce welfare.

Our paper explores the observation that in software markets, both strong and weak protection of intellectual property may foster innovation, depending on the characteristics of the software market at hand. Thus, sometimes it may make sense to stimulate OSS, while in other cases, PS works better for innovation. We discuss innovation incentives and market failures in software markets, and based on this, discuss what policy makers can do to alleviate market failures.

---

4 Ministry of Economic Affairs (2007), p. 6.
For a formal definition see http://www.opensource.org/docs/osd.

The relevance of our paper is illustrated by various policy initiatives [5]. For example, since 2007, the Dutch government has actively stimulated the adoption of OSS in the public sector [6]. France has been considering the promotion of OSS by public administrations at least since 1999, although it took longer before concrete initiatives were adopted. Currently, the Gendarmerie Nationale is a prominent example of an organization that switched to OSS on a large scale. In Spain, a plan was adopted in 2003 that called for the adoption of OSS when available and adequate. The broad scope of proposals observed in practice demonstrates the relevance of pinpointing the economic rationale behind public support for OSS.

The substance of our arguments comes from recent literature on innovation. An important reference is Bessen and Maskin (forthcoming), who argue in favor of weak IPR protection when innovation is interactive and sequential. There exist various papers related to ours, that contain policy analyses of the relative merits of OSS and PS [7]. Whereas SCHMIDT & SCHNITZER (2003) recommend against any policy intervention that directly promotes OSS, we argue that depending on the market failures that may occur, intervention may be desirable. MAURER & SCOTCHMER (2006) argue that "[…] open source incentives reduce agency problems and deadweight loss compared to patents, and accelerate discovery through automatic disclosure [but] often lead to an under-supply of goods [and] may also be less responsive to certain users, especially when those users are non-programmers" (p. 313). Further, as LEE (2006) states, the "policy considerations that inform government decisions are extremely complicated and sometimes interdependent" (p. 112). The complexity stems from the technological aspects related to various forms and combinations of market failures that may occur in specific markets – they are not uniform for the software market as a whole.

Note that our paper does not address anticompetitive behaviour in software markets, for instance related to bundling and exclusionary behaviour. Our focus is on situations in which a supplier may have a dominant position, in particular due to customer lock-in, but is not abusing

5 See http://www.osor.eu, the website of the Open Source Observatory and Repository for European public administrations (OSOR).

6 “The Netherlands in Open Connection”, Ministry of Economic Affairs (2007).

7 MENDYS-KAMPHORST (2002) analyzes the impact of open source projects on competition. SCHWARZ & TAKHTEYEV (2009) relate the success of open source software to the fear of holdup.

that position. In such cases, competition law enforcement has no bite. Nevertheless, public policy may be able to increase welfare.

The remainder of this paper is as follows. The next section discusses development incentives and market failures in software markets. The subsequent section discusses the role of the government. The conclusion recapitulates our findings.

■ Characteristics of open source and proprietary software

Incentives to participate in OSS development [8]

Why do individual software developers (not employed by firms) voluntarily contribute to OSS projects? A programmer, when participating in software development, faces a variety of private benefits and costs[9]. We assume that he or she decides to participate in a project if the benefits outweigh the costs.

PS developers employed by firms receive direct payments in the form of wages or performance-contingent rewards. Independent OSS programmers do not necessarily receive such payments, while the nature of OSS licenses, to some extent, makes selling the software more difficult. Therefore, they must be driven by incentives to exert effort other than direct monetary payments. These incentives may be both intrinsic and extrinsic in nature[10]. The most relevant extrinsic motivations are: (i) benefits from own use, (ii) job market signalling (peer recognition and professional status enhancement), and (iii) self-education (skill enhancement). Note that (ii) and (iii) may also be motivations for programmers working on PS, and may relate to future monetary pay-offs. An intrinsically motivated person derives utility from an activity itself, or from psychological aspects directly related to it (such as altruism, meritocracy or reciprocity).

---

[8] Based on LERNER & TIROLE (2005a, 2005b), MAURER & SCOTCHMER (2006) and MENDYS-KAMPHORST (2002).

[9] Costs include the opportunity cost of time, such as the forgone compensation from another task or the value of leisure time.

[10] See also HANN *et al*. (2004) and LAKHANI & WOLF (2005).

Why do profit-maximizing firms develop OSS? A software-developing firm may generate revenues from selling the software or related complementary products and services. It faces initial software development costs. These costs are substantial when development, besides programming skills, requires expensive, specific inputs, such as scarce expert knowledge. After this initial investment cost has been incurred, the incremental production and sales costs are small, as copies are easily made.

The expected financial benefits of developing OSS should outweigh the costs of giving up the profits that arise from selling the software under a proprietary license. Open source projects initiated by private firms should therefore meet two requirements. Firstly, implementation of the project should be feasible. Private firms developing OSS face several challenges in this respect. For instance, it helps if the project can be divided into separate modules on which programmers can work independently, so that the innovation process benefits from open interaction. Also, the firm must be able to commit to keep the source code disclosed, for example by surrendering control over the project to outside parties, or by choosing a licence that forbids the software to be reverted to PS (e.g. General Public License or GPL). Secondly, the open source project must be economically viable. OSS developers do not — unlike PS developers, who have the exclusive right to market their products by selling their rights or licensing to others — directly derive market power from their license[11]. Everybody can use, improve and distribute the software for free. Thus, OSS has to be profitable for other reasons. One possible business model is what Lerner and Tirole (2005a) call "living symbiotically". OSS may generate revenues in complementary and proprietary activities, such as support, education, training, and complementary PS. Another rationale to develop OSS may arise when a firm is lagging behind a PS supplier but sees the possibility that its product will be successful as OSS, thereby increasing profitability in complementary segments (LERNER & TIROLE, 2005a). Open source development may also counter-weigh the dominant position of a competitor, or it may increase reputation and visibility in the long run (HENKEL, 2005). A final source of revenues may stem from reduced development costs. Firms can pay employees lower wages because they are intrinsically motivated. Development costs may be lower because of external development support, debugging and software maintenance by OSS communities.

[11] We define market power as the ability to charge a price above marginal cost (a mark-up).

Market failures in software markets [12]

There is a variety of potential market failures in software markets. Market failures result in sub-optimal prices, quality or innovation levels. For policy makers it is therefore important to understand potential market failures and the ways in which they affect welfare. Software markets typically exhibit the following market failures: knowledge spillovers, market power, and economies of scale.

Knowledge spillovers occur if firms benefit from knowledge creation by others. We distinguish between imitative spillovers (slightly differentiated products based on an existing technology) and creative spillovers (follow-on innovations that extend the original technology to a significant extent). Imitative spillovers reduce an innovator's profits and hence reduce the *ex ante* incentives to innovate. Creative spillovers arise when the innovation process is interactive, so that developers improve on each other's contributions. If a firm does not take into account the additional benefits due to creative spillovers when making investment decisions, there may be too little investment from a social viewpoint. This effect is particularly pronounced when upfront development costs are large: not taking into account spillovers may then have the effect that a socially desirable investment that is privately unprofitable, is not carried out.

Market power, another type of market failure, is in software markets often caused by customer lock-in. This phenomenon typically occurs when there are network effects or switching costs. Network effects arise when the utility that a user derives from consumption increases with the number of other users of the same (or a compatible) product [13] [14]. One can distinguish two types of network effects. Direct network effects arise when consumers value a particular piece of software more when there are more other users of the same or compatible software (office applications are an example). Indirect network effects arise when consumers value a particular software program

12 This sub-section draws on FARRELL & KLEMPERER (2007), SCHMIDT & SCHNITZER (2003), SHY (2001) and MAURER & SCOTCHMER (2006).

13 This is the definition from KATZ & SHAPIRO (1985), who use the term “network externalities”, just as most of the subsequent literature. LIEBOWITZ & MARGOLIS (1994) argue that this should better be called a network effect. They reserve the term "network externality" for network effects in combination with unexploited gains from trade with regard to network participation. Their aim is to make precise when the presence of network externalities leads to welfare distortions.

14 Network effects can be present both on the demand side and the supply side of software markets. For our purposes, we abstract from supply-side effects.

more when a larger number of complementary applications exist (e.g. in the case of PC operating systems) [15]. When users cannot communicate with the majority of other users, network effects may complicate or distort consumers' decisions to choose between incompatible programs. Thus, a market failure may occur when users cannot coordinate the transition from one equilibrium outcome to another one. An example is a switch from operating system Windows to Linux, or from the Microsoft office application suit to OpenOffice). When such a coordination problem occurs, some users may get stuck with the program that has the largest customer base but may nevertheless be a second choice regarding quality or specifications.

Switching costs, another cause of lock-in, arise when consumers incur costs when they switch to another technology, product or supplier. Switching costs for instance occur because of the need to learn how to use a different product, to buy new compatible software programs [16] or to search for an alternative product. ZWIEBEL (1995) provides an alternative rationale for switching cost, arguing that managers, to protect their reputation or because of a lack of information, may abstain from switching to an alternative solution that is superior to the industry standard or the product of a dominant, well-known supplier.

Economies of scale form another potential source of market failure. Software markets are characterized by economies of scale because the development of the "first copy" of a product entails a fixed cost, while the costs of producing additional copies are close to zero. Scale economies are particularly pronounced for software requiring a costly input of expertise complementary to programming skills, that is painstakingly developed by specialists other than software developers (and can easily be copied). Think, for instance, of expertise related to complicated econometric methods (as in the case of advanced statistical software), specific accounting and business administration skills, well-trained user support (as in the case of business administration software), or graphic designer skills (as in the case of high-end games). The presence of scale economies makes it necessary that suppliers charge a price above marginal cost, in order to remain viable.

---

[15] One could add here the presence of 'applications barriers to entry', which occur, for instance, if no one can compete with an existing operating system without having a sufficient number of applications running on it. To some extent, this situation may be seen as a case of indirect network effects, or alternatively, as a two-sided market (ROCHET & TIROLE, 2006).

[16] Close standards, for instance, restrict the exchange of files between users of different applications, therefore making switching costly.

■ The potential role of the government

If market failures exist, government intervention may be able to improve market outcomes in terms of welfare. We will discuss which roles the government could take up and how they relate to the potential market failures that may occur [17]. Attention will also be given to the potential negative side effects of government intervention, that is, government failure.

Government policy

IPRs address knowledge spillovers: (i) they provide protection against imitation of an innovator's idea, and (ii) they allow the innovator to appropriate surplus from third-party follow-on innovations through licensing. A proprietary license is a common way to make use of the protective power of IPRs. Accordingly, through the prospects of financial recoupment, a proprietary license safeguards the incentives to exert development efforts. In the absence of proprietary licenses, software with substantial, initial development costs would perhaps not be developed. Such types of software are therefore best brought to the market as PS. Furthermore, PS tends to cater for the tastes of the "average" consumer and therefore has a larger potential market. The reason is that PS development is driven by expected profits, which is affected by consumers' willingness to pay, and thus by the benefits consumers expect to derive from it (See also SCHMIDT & SCHNITZER, 2003) [18]. Therefore, a profit maximizing firm will prefer to invest in software that appeals to a broad group of "unsophisticated" users (who are not programmers by profession). In the absence of proprietary licenses, innovation, particularly in the initial stage where consumers' needs do not yet play an important role, would be primarily driven by the motivation of individual software developers themselves (MAURER & SCOTCHMER, 2006) [19]. These incentives do not take into account the needs of an average user. Thus, OS development provides fewer incentives to appeal to a broad group of users. Note that without commercial motives, certain ideas get a

---

[17] It is outside the scope of this paper to compare the effectiveness and the pros and cons of the different policy options that are available.

[18] Note, however, as VON HIPPEL (2002) argues, it is sometimes costly for firms to follow heterogeneous consumer needs, for example because it is hard to know individual consumers' preferences or it is impossible to differentiate prices.

[19] For commercial open source projects, e.g. in which profits are meant to be generated through complementary products and services, this argument does not apply.

chance to be developed that would not stand a chance to be adopted in a commercial environment. At a later stage, it may turn out that a fraction of these ideas do lend themselves for commercial exploitation. Thus, OSS may help to "let a thousand flowers bloom", some of which may turn out to become mass market products at a later stage.

It is important to note that IPRs introduces its own distortions. First, the market power derived from IPRs may be strengthened by network externalities, switching costs, and economies of scale. Market power may therefore be substantially higher than necessary to safeguard innovation incentives. This comes at a cost for welfare: (i) mark-ups create a deadweight loss resulting from reduced demand [20]; (ii) market power may facilitate anti-competitive behaviour, especially when customer lock-in effectively results in an entry barrier. Second, the appropriation argument above assumes that an innovator and a producer can contract efficiently. However, efficient contracting may not be possible due to information asymmetries, or if contracts are incomplete. Information asymmetries may, for instance, arise when the contribution of follow-on innovators is unpredictable, or if the value of potential contributions is private knowledge of contributors. Competitors who want to build on an existing innovation may have useful ideas about how such innovations can be achieved, that are not available to the original innovator. In that case, proprietary licenses may slow down the speed of innovation relative to the socially optimal level of innovation. This is essentially the approach taken by Bessen and Maskin (forthcoming), who develop a model of sequential innovation with private information about development costs. They note that the welfare loss that occurs when an innovator is unable to appropriate the surplus from follow-on innovations, need not occur when IPRs are weak (or under open source licensing). When a software license allows for the use of the source code by others, follow-on innovations can take place without hindrance. In their model, whether weaker or stronger protection is optimal depends on the specific details of the distribution of the social value generated by innovations. They find that weak protection is optimal if valuable follow-on innovations are sufficiently likely. Thus, if efficient contracting is not possible, certain socially valuable follow-on innovations will not emerge under proprietary licensing, while they would occur under open source licensing. Therefore, if there is no need to purchase expensive, specific inputs from

20 The deadweight loss is reduced if firms can engage in price discrimination, and thereby increase the number of users of their products.

specialists outside the software community, open source licenses result in higher social welfare [21].

Thus, there exists a fundamental trade-off between *ex ante* incentives to innovate and *ex post* efficiency. While IPRs are a response to knowledge spillovers, they may introduce new distortions, by creating market power or slowing down the innovation process. Now, when consumers can easily switch to other suppliers, this will not be a problem, since superior technologies will then be able to establish a position in the market. However, when there is a serious problem of customer lock-in, this may be problematic. In such cases, it may be worthwhile to stimulate OSS in order to overcome customer lock-in, that is, to alleviate the coordination failure in the market. An alternative solution, when network externalities are present, is to guarantee a seamless exchange of files between users, for instance through enforcement of open standards.

Whether intervention is desirable depends on which licensing regime dominates from a welfare perspective, and which market outcome can be expected to arise in the absence of government intervention. To derive policy recommendations we will assess whether stimulating OSS creates downward price pressure (and improves "static efficiency"), or that, in addition, it impacts the endogenous emergence of technologies and efficiently results in more innovation (and improves "dynamic efficiency") [22]. There may exist a trade-off static efficiency and dynamic efficiency, located on the right-hand side of the inversed U-shaped curve depicting the relationship between competition and innovation (AGHION & GRIFFITH, 2005). In particular, it is possible that intensified competition reduces innovation incentives, for instance if the market is already competitive, so that further increasing the intensity of competition would lead to cut-throat competition, by eroding the margins needed to invest in innovation. However, a serious problem of customer lock-in typically results in a low intensity of competition to start with. In such a situation, increasing competition tends to go hand in hand with increasing the incentives for innovation, and will therefore increase dynamic efficiency (a move to the right on the left-hand side of the inverse U-shaped curve).

---

21 The incentives provided by PS may co-exist with some of the advantages of OSS, especially when there are informal or enforced standards to ensure interoperability and compatibility between applications and file formats.

22 See BENNETT *et al.* (2001) for definitions of static and dynamic efficiency.

Now consider a market situation in which PS dominates, while OSS plays a minor role. This is a natural point of departure to assess whether it makes sense to stimulate OSS. In the short run, stimulating OSS typically increases the intensity of competition, since it introduces downward price pressure on proprietary products [23]. Software users will not be willing to pay high prices if a comparable product can be obtained for free. In addition, when there is customer lock-in, stimulating OSS, for instance by acting as a lead customer, may help to overcome entry barriers, thus further increasing static efficiency. Stimulating OSS is therefore, in general, good for static efficiency. Whether or not stimulating OSS increases or decreases dynamic efficiency is pivotal in justifying government policy, because long-term welfare effects are typically of a higher order of magnitude. However, the effect of stimulating OSS on dynamic efficiency is ambiguous and more difficult to assess. For software requiring expertise complementary to programming skills at a substantial cost, a business model based on PS tends to lead to more innovation, since it is more suitable to get development off the ground. Stimulating OSS may then reduce dynamic efficiency. Moreover, if efficient contracting on follow-on innovation is possible, developers with market power can provide (efficient) firms that wanted to develop follow-on innovation with the source code, in return for (a part of) the profits that these new products will generate. If this is the case, stimulating OSS will not lead to more follow-on innovations than the efficient level. Nevertheless, if efficient contracting of proprietary source code is not feasible, then stimulating OSS will lead to a higher and more efficient level of innovation [24].

Based on the previous discussion, we suggest the following policy guidelines. Policy makers should first consider whether serious customer lock-in is present in the market. If so, it has to be assessed what type of business model, if any, is the most appropriate in terms of dynamic efficiency and in the light of the other market failures that may occur in software markets.

Suppose first that there is no serious customer lock-in problem. Should the government then intervene in order to stimulate OSS? At the demand side, customers do not face serious hurdles to switch to new entrants. At the

[23] This depends on the extent to which OSS and PS offerings are substitutes. For instance, for a long time, Linux was not been seen as a fully fledged substitute for Windows.

[24] To avoid broad-brush sponsoring of a wide range of pet projects, with all the inefficiencies that go with it, a potential need for government intervention arises only at a later stage, when markets for some of the “thousand flowers” have already come into existence.

supply side, entrants (whether they offer PS or OSS) do not face substantial entry barriers impeding them from competing with an existing software supplier [25]. When such entrants are more efficient or innovative, they will be able to gain market share quickly. Market processes are therefore likely to result in an efficient constellation of software offerings and types, and there is no need to interfere with the market-driven emergence of innovations. Abstaining from *ex ante* intervention and, if necessary, relying on competition law enforcement, is then the best option to avoid distorting market processes (we will come back to this).

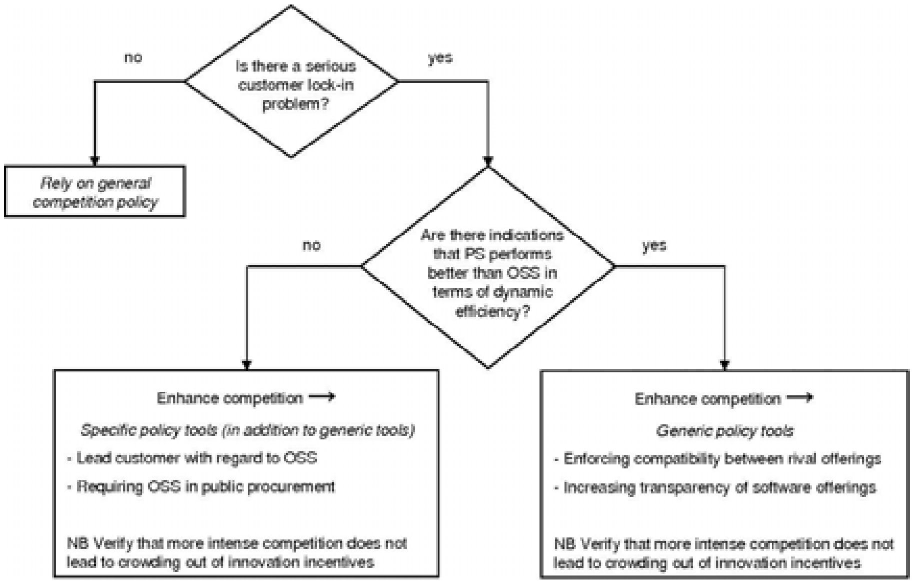
Second, suppose that there is a serious problem of customer lock-in. Then, in a market dominated by an incumbent offering PS, new entrants may have too little chance to enter, even if they offer a superior product at a lower price. Now the question is whether there are indications suggesting that, from a welfare perspective, PS is a superior business model compared to OSS. As we discussed, an important point is that from a welfare viewpoint, OSS is not the best business model in all circumstances. We identified two conditions that determine whether OSS or PS business models are optimal: (i) software development may require a specific, complementary input at a substantial cost, and (ii) there may be no obstacles with regard to efficient contracting on follow-on innovations. If both conditions are satisfied, this indicates that PS performs better than OSS in terms of dynamic efficiency. It may then not be wise to introduce policy specifically aiming at stimulating OSS. Actively stimulating OSS in such markets conflicts with the effectiveness of PS-based business models, and can lead to distortions of dynamic efficiency. However, note that generic policy to reduce customer lock-in (e.g. by enforcing open standards and seamless exchange of files) could be desirable. If only condition (i) holds, policymakers have to assess that stimulating OSS does not hurt firms' incentives to innovate. If only condition (ii) holds, then stimulating OSS will probably not efficiently increase the level of innovation. If neither condition holds in a specific software market, then, besides enhancing competition by generic instruments, directly stimulating OSS by specific policy may be very useful to help the market overcome customer lock-in. We discuss generic and specific policies below.

One can now construct the following decision tree for policy makers.

---

25 We abstract from capital market imperfections.

Decision tree for policy makers



Let us apply this framework to some tentative examples, meant to provide an illustration and first intuition on how to use our framework [26].

- In the market of advanced statistical and mathematical software, lock-in problems are less substantial than in markets where customers exchange files and applications on a continuous basis (e.g. operating systems and office applications). Hence network externalities form no substantial barriers to enter the market. Other entry barriers may of course exist, but those are likely to be less problematic.

- Also in the case of advanced statistical software, development costs are high due to the need of specific, complementary inputs, that is, experts who understand complicated econometric methods and statistical procedures. Therefore PS may be more appropriate to provide for cost recoupment possibilities. Nevertheless, an entrant offering a superior alternative will be able to take over the incumbent's position and recoup the necessary investment. Examples in which the development of software does not (or to a lesser extent) require complementary, specific inputs, are internet browsers and web content management software.

[26] The examples are not meant as a call for policy measures – further analysis is needed in specific cases.

- In the case of high-end interactive games, in which players spend significant amounts of time playing with each other, network effects are strong, creating entry barriers. At the same time, the need to purchase advanced graphical skills makes the development costs high. Similarly, there may be high switching costs in the market of enterprise resource planning software. Development of business administration software may require specific inputs, like specific accounting and business administration skills, that increase development costs significantly. In these cases, PS-based business models may do a better job in cost recovery.

- PC operating systems also exhibit substantial (indirect) network externalities and switching costs. However there is relatively less need for specific, complementary expertise (beyond adequate programming skills) to develop such software. Thus there is no indication that PS performs better than OSS in stimulating innovation. By stimulating OSS in a way that creates a significant installed base for rival offerings, consumers' barrier to switch can be reduced substantially. Other examples where this argument applies are office software (text editors, type setting applications, spreadsheets, and presentation editors) and enterprise content management software.

One can distinguish between generic and specific policy tools. We call a policy tool generic if it addresses a market failure independently of the type of software (e.g. characterized by the license form). When policy intervention targets OSS, we classify it as specific.

Let us first discuss generic policy tools. Firstly, facilitating or mandating interoperability (in the sense of compatibility) between competing products can help to internalize network externalities by allowing consumers of different software programs to exchange files in a seamless manner (see also LEE, 2006). This type of intervention may be particularly important when PS suppliers have incentives to hinder interoperability, for example by implementing closed standards in order to preserve their market power, or by fiddling with the specifications that files generated by competing software have to comply to. Secondly, to lower the perceived (non-financial) switching costs of consumers, the government could improve transparency of software offerings by facilitating the provision of information about the availability, price and specifications of products and relating services. Note that these types of *ex ante* policy intervention may also involve, to a certain extent, the application of competition law (*ex post* intervention), with the purpose of keeping the market open for innovation and ensuring that ultimately consumers decide which standard, technology or product will win [27].

We now turn to specific policy tools. The government can stimulate OSS by acting as a lead customer, for instance by promoting or even requiring the use of OSS in the public and semi-public sector or by imposing OSS as a requirement in public procurement. Specific policy may be effective in "tipping" the market towards a viable market share for OSS. The underlying mechanism is that by doing so, the government creates a critical mass of users of an OSS alternative, making it more attractive for other consumers to follow. Note that large corporate clients may also more easily adopt OSS if purchasing managers can motivate their choice for OSS by pointing to the fact that well-established organizations also purchase OSS (ZWIEBEL, 1995).

The set of policy tools discussed here is not exhaustive but rather an overview of relatively light-handed types of intervention. In particular, our shortlist does not include demand subsidies (e.g. government sponsored distribution of OSS amongst citizens, teaching in schools) or supply subsidies (e.g. tax subsidies for software developers developing OSS, annual prizes for the best OSS product).

Government failures

It is important to acknowledge that good intentions are not sufficient to motivate government interventions, since they can have benefits as well as costs. Accordingly, to be able to conclude that a policy indeed corrects a market failure, one must ask whether it actually reduces the inefficiency or welfare loss that was caused by the market failure. In other words, one must take into account the possibility of government failures, which are failures that arise when the government introduces a new inefficiency "because it should not have intervened in the first place or when it could have solved a given problem [...] more efficiently, that is, by generating greater net benefits" (WINSTON, 2006, pp. 2-3).

Unfortunately, unlike the standard list of potential market failures on which economists agree, there is no received wisdom with regard to potential government failures. Ultimately, the effects of intervention can only

[27] For more on this in the light of the European Microsoft court case, see LAROUCHE (2008).

be assessed empirically. For the purposes of this paper, however, it is useful to consider three important types of failure that should be taken into consideration before implementing a specific policy.

Firstly, when the government acts as a lead customer or steers the market in other ways, there is no guarantee that it makes a choice that is not inferior to choices made at a decentralized level by market participants. This type of argument is a standard criticism in the economic literature on innovation policy, where economists usually recommend against policies that aim to pick or support winners (BOONE & VAN DAMME, 2004). As SCHMIDT & SCHNITZER (2003) argue in the context of software markets, independently of the presence of market failures, an inappropriate product choice could tip the market in the wrong direction, and by doing so, competition and innovation incentives may be reduced.

Secondly, policy makers may not be able to assess properly the long-term success of a publicly supported OSS project, for instance due to incomplete information. As a result of public procurement biased towards OSS, a currently profitable and efficient provider of PS may then lose its incentives to invest.

Thirdly, government procurement processes based on a bias towards a certain type of technology, which involves making choices between different projects or suppliers, invites lobbying by candidate suppliers, and typically introduces inefficiencies due to rent-seeking behaviour.

■ Conclusions

The important market failures in software markets are market power, due to network externalities, switching costs or economies of scale, and knowledge spillovers. Software developers may be driven by monetary as well as non-monetary incentives. When intervening in software markets, policy makers have to take into consideration that there may exist a trade-off between reducing market power and encouraging follow-on innovations on the one hand, and preventing firms from free riding on other firms' efforts to innovate on the other. In other words, there may exist a trade-off between *ex post* efficiency and *ex ante* incentives. Whether this trade-off exists or not, determines what business model is optimal from a welfare perspective. If so, PS tends to be optimal, if not, then OSS has strong advantages. More precisely, we pointed out that proprietary business models tend to perform better in terms of innovation if development costs are high due to the need of costly, specific inputs and when optimal contracting on license fee about derived works is efficiently feasible. In contrast, open source business models tend to dominate proprietary business models, if there is no need to purchase expensive inputs complementary to programming skills, creative spillovers are socially valuable, and proprietary licenses are too restrictive to provide sufficient incentives for follow-on innovations. Note that these are stylized arguments – specific cases require separate, more detailed studies.

Before discussing the policy recommendations, let us stress that specific instead of generic policy options require a strong motivation, because of the risk of government failure. In practice, competition policy and making sure that consumers can "vote with their feet" are the best candidates to counterbalance market power and reduce entry barriers. The reason is that consumers are better informed about their own preferences with regard to new products than the government.

The extent to which *ex ante* intervention in addition to general competition policy is advisable depends on the presence of market failures. First of all it has to be assessed whether there are serious customer lock-in problems in the market. If so, policy intervention may be recommendable, depending on which business model better safeguards or stimulates innovation incentives. If proprietary business models perform better, generic policy options (not specifically aiming at OSS) may reduce customer lock-in and lower entry barriers. Examples of generic policy options are requiring interoperability or compatibility, and increasing transparency of software offerings. Stimulating OSS can be recommendable when open source business models dominate in terms of innovation incentives. Specific policy options that can help to alleviate customer lock-in problems and increase dynamic efficiency are, for instance, acting as a lead customer or requiring OSS in public procurement.

References

AGHION, P. & R. GRIFFITH (2005): *Competition and Growth: Reconciling Theory and Evidence*, MIT Press.

BENNETT, M., P.J.W. DE BIJL & M. CANOY (2001): "Future Policy in Telecommunications, An Analytical Framework", CPB Document 5.

BESSEN, J. & E. MASKIN:

- (2004): "Intellectual Property on the Internet: What's Wrong with Conventional Wisdom?" Available at: http://www.researchoninnovation.org/iippap2.pdf.

- (forthcoming): "Sequential Innovation, Patents, and Imitation", *RAND Journal of Economics*. Available at: http://www.researchoninnovation.org/patrev.pdf.

BIJLSMA, M.J., P.J.W. DE BIJL & V. KOCSIS (2009): "Competition, innovation and intellectual property rights in software markets", CPB Document 181.

BOLDRIN, M. & D.K. LEVINE (2009): "Intellectual Property Rights and Economic Growth in the Long Run: A Model of Discovery", *American Economic Review: Papers & Proceedings* 99(2), pp. 337–342.

BOONE, J. & E.E.C. VAN DAMME (2004): "Marktstructuur en innovatie", In: B. JACOBS & J.J.M. THEEUWES (Eds), *Innovatie in Nederland: De Markt Draalt en de Overheid Faalt*, KVS Amsterdam, pp. 71-92.

GALLINI, N. & S. SCOTCHMER (2002): "Intellectual Property: When Is It the Best Incentive System?", in: A. JAFFE, J. LERNER AND S. STERN (Eds), *Innovation Policy and the Economy*, Vol. 2, MIT Press, pp. 51-78.

GUIBAULT, L. & O. VAN DAALEN (2005): "Unravelling the Myth around Open Source Licenses: An Analysis from a Dutch and European Law Perspective", Mimeo, Institute for Information Law, University of Amsterdam.

FARRELL, J. & P. KLEMPERER (2007): "Coordination and Lock-In: Competition with Switching Costs and Network Effects", in: M. ARMSTRONG & R. PORTER (Eds), *Handbook of Industrial Organization*, Vol. 3, Elsevier, pp. 1967-2072.

HANN, I-H., J. ROBERTS, S. SLAUGHTER & R. FIELDING (2004): "An empirical analysis of economic returns to open source participation", Mimeo, Carnegie Mellon University.

KATZ, M.L. & C. SHAPIRO (1985): "Network Externalities, Competition, and Compatibility", *American Economic Review* 75(3), pp. 424-440.

LAKHANI, K.R. & R.G. WOLF (2005): "Why hackers do what they do: understanding motivation and effort in free/open source software projects", in: J. FELLER, B. FITZGERALD, S. HISSAM & K.R. LAKHANI (Eds), *Perspectives on Free and Open Source Software*, MIT Press, Cambridge, MA, pp. 3-21.

LAROUCHE, P. (2008): "The European Microsoft Case at the Crossroads of Competition Policy and Innovation", TILEC Discussion Paper 2008-021, Tilburg University. Available at http://ssrn.com/abstract=1140165.

LEE, J.-A. (2006): "New Perspectives on Public Goods Production: Policy Implications of Open Source Software", *Vanderbilt Journal of Entertainment and Technology Law*, 9(1), pp. 45-112.

LERNER, J. (2009): "The Empirical Impact of Intellectual Property Rights on Innovation: Puzzles and Clues", *American Economic Review: Papers & Proceedings* 99(2), 343-348.

LERNER, J. & J. TIROLE:

- (2005a): "The Economics of Technology Sharing: Open Source and Beyond", *Journal of Economic Perspectives*, 19(2), pp. 99-120.

-(2005b): "The Scope of Open Source Licensing", *Journal of Law, Economics, and Organization*, 21(1), pp. 20-56.

LIEBOWITZ, S.J. & S.E. MARGOLIS (1994): "Network Externality: An Uncommon Tragedy", *Journal of Economic Perspectives* 8(2), pp. 133-150.

MAURER, S.M. & S. SCOTCHMER (2006): "Open Source Software: The New Intellectual Property Paradigm", in: T. HENDERSHOTT (Ed.), *Handbook of Economics and Information Systems*, Elsevier, pp. 285-319.

McGOWAN, D. (2005): "Legal Aspects of Free and Open Source Software", in: J. FELLER (Ed.), *Perspectives on Free and Open Source Software*, MIT Press, pp. 439-440.

MENDYS-KAMPHORST, E. (2002): "Open vs. closed: Some consequences of the open source movement for software market"s, CPB Discussion Paper 13.

Ministry of Economic Affairs (2007): "The Netherlands in Open Connection: An action plan for the use of Open Standards and Open Source Software in the public¬ and semi-public sector", The Hague.
Available at http://appz.ez.nl/publicaties/pdfs/07ET15.pdf.

ROCHET, J-C. & J. TIROLE (2006): "Two-Sided Markets: A Progress Report", *RAND Journal of Economics*, 35(3), pp. 645-667.

SCHWARZ, M. & Y. TAKHTEYEV (2009): "Half a Century of Public Software Institutions: Open Source as a Solution to Hold-Up Problem", NBER Working Paper 14946.

SHY, O. (2001): *The Economics of Network Industries*, Cambridge University Press.

SCHMIDT, K. & M. SCHNITZER (2003): "Public Subsidies for Open Source? Some Economic Policy Issues of the Software Market", CEPR Discussion Paper 3793.

STIGLITZ, J.E. (forthcoming): "Economic Foundations of Intellectual Property Rights", *Duke Law Journal*.

VARIAN, H. (1989): "Price Discrimination", in: R. SCHMALENSEE & R. WILLIG (Eds), *Handbook of Industrial Organization*, Vol. 1, Amsterdam, North-Holland, pp. 597-654.

VON HIPPEL, E (2002): "Open source software projects as user innovation networks", Mimeo, MIT Sloan School of Management.

WINSTON, C. (2006): *Government Failure versus Market Failure. Microeconomics Policy Research and Government Performance*, AEI – Brookings Joint Center for Regulatory Studies, Washington, D.C.

ZWIEBEL, J. (1995): "Corporate Conservatism and Relative Compensation", *Journal of Political Economy* 103(1), pp. 1-25.