Coexistence and Market Tipping in a Diffusion Model of Open Source vs. Proprietary Software*

Riccardo Leoncini†
Francesco Rentocchini‡
Giuseppe Vittucci Marzetti§

April 7, 2009

Abstract

Although open source software has recently attracted a relevant body of economic literature, a formal treatment of the process of competition with its proprietary counterpart is still missing. Starting from an epidemic model of innovation diffusion, we try to fill this gap. We propose a model where the two competing software depend on different factors, each one specific to its own mode of production (profits and developers' motivations), together with network effects and switching costs. As the speed of diffusion can affect the final outcome, we endogenize the parameter influencing it across the population of adopters. We find that an asymptotically stable equilibrium where both software coexist can always be present and, when the propagation coefficient is endogenous, it coexists with winner-take-all solutions. Furthermore, an increase in the level of the switching costs for one software increases the number of its adopters, while reducing that of the other one. If the negative network effects increase for one of the two software, then the equilibrium level of users of that software decrease.

Keywords: Increasing returns; Innovation diffusion models; Open-source software; Technological competition.

JEL Classification: L17; L86; O33.

---

*Previous versions of the paper were presented to the PRIN workshop on "Fragmentation and local development: interpretative models and policy implications", University of Padua, 3-4 June 2008, and to the EAEPE Conference, Rome, 6-8 November 2008. The authors gratefully acknowledge comments and suggestions from the participants, and in particular Giorgio Negroni, Marika Macchi and Ulrich Witt. The usual caveat applies. Francesco Rentocchini and Giuseppe Vittucci Marzetti gratefully acknowledge the financial support from the Autonomous Province of Trento (Grants 2006: TRACKs and FIC).

†Dept. of Economics, University of Bologna, riccardo.leoncini@unibo.it

‡Dept. of Economics, University of Trento, francesco.rentocchini@economia.unitn.it

§Dept. of Economics, University of Trento, giuseppe.vittucci@economia.unitn.it

1 Introduction

The growth of adoption of Open Source Software (OSS) in recent years has attracted the attention of many scholars from different fields. A large number of case studies has been carried out to explain and empirically ground such a relevant gain in popularity. Among the others, topics such as the organisation and ethos of the community of developers, together with their motivation to provide code for free and the birth of hybrid business models have been extensively examined by different branches of literature.

An issue that has not been sufficiently investigated so far and that needs particular attention is the one pertaining to the OSS innovation model and the way it competes against the Proprietary Software (PS) one. Few analyses have coped with the issue, and they usually lack a formal treatment. To our knowledge, only few contributions have tried to overcome such a shortfall (e.g. Bonaccorsi and Rossi, 2003; Dalle and Jullien, 2003), though they rely on quite *ad hoc* models, rather than on the more developed models of innovation diffusion within the literature of innovation economics.

Our starting point is thus to model the process of competition between OSS and PS through an epidemic model able to take into account both demand and supply side factors. We make the two technologies depend on different sets of determinants. In particular, OSS fundamentals depend on a set of factors at odd with the usual ones (Lerner and Tirole, 2002), such as reputational and communitarian factors (intrinsic and extrinsic motivations), while those of PS are essentially profit motivated.[1]

In particular, we will use an epidemic model, as it fits better the process of diffusion of software packages, which is affected by the size of the installed base.[2] Indeed, it is quite straightforward for this class of models to depict within-technology network effects as the result of contagion (due, for instance, to word-of-mouth interactions).[3] Furthermore, epidemic models are better suited than game theory models to describe situations in which, on the one side, there are important problems related to the effects of the transmission of knowledge among agents about the unknown characteristics of a technology, and, on the other side, we are less interested in strategic

[1]Nonetheless, a considerable number of large firms decided to enter into the software market in order to benefit from this process. Among the others, IBM, Novell and Dell are worth mentioning. This fact has led to the creation of a new type of hybrid business model characterised by the presence of for-profit companies benefiting from OSS solutions, mainly developed by the not-for-profit OSS communities they support (West, 2003; Fosfuri *et al.*, 2008). Furthermore, an increasing number of countries all over the world has started discussing about the role OSS should have in public administration.

[2]An increasing number of studies provides empirical evidence on the importance of network effects, both direct and indirect, in different market segments of the software industry. See, for istance, Gandal (1994) and Brynjolfsson *et al.* (1996) for spreadsheets; Chiaravutthi (2006) for browsers; and Gandal (1995) for PC software.

[3]As it will be clear in the following, for the across-technology network effects we will introduce interoperability issues between the two technologies.

behaviour, since OSS developers cannot "by definition" take into account how the evolution of their software will impact upon the competing one. Hence, it is easier to consider the "strategic" behaviour of only one side (the PS) considering that some of its possible choices are unrelated to the competitor's reaction.

At the same time, both PS and OSS diffusion are influenced by the presence of network effects, the level of interoperability, and the presence of switching costs. Finally, compared to standard diffusion models, we endogenise the parameter influencing the speed of diffusion across the population of adopters. In so doing, from a theoretical perspective, our model offers a formal treatment to solve more complex and more realistic diffusion models; from a more practical perspective, it gives a way to properly deal with diffusion patterns whose speed can vary along the process since the interactions among the technologies contribute to change the incentives for consumers as they proceed in the discovery process of their characteristics.

The paper is structured as follows. In Section 2, we present the theoretical background. In particular, available results in the area of both competition between proprietary and open source software (Section 2.1) and innovation diffusion models (Section 2.2) are put forward. On the grounds of the previously mentioned results, we then discuss the general framework within which the two competing technologies must be conceived, that is the software industry. In particular, a set of three main features are considered: network effects, interoperability and switching costs. These characteristics are then incorporated in the formal model developed in Section 3. The results are then presented in Section 4. Several variants of the model are discussed: (i) a base version characterised by constant propagation coefficients for the two technologies (Section 4.1); (ii) an extended version with changing propagation coefficients with (Section 4.2.2) and without network effects (Section 4.2.1). Section 5 concludes.

2 Theoretical background

2.1 The competition between Proprietary and Open Source Software

The issue of OSS has gained momentum in recent years thanks to the echo derived from a number of relevant "success stories". Firefox among internet browsers, Apache among web servers, OpenOffice among office suites and Fetchmail among the mail sending protocols are all well known examples.[4] This increasing popularity can be explained by several stylised facts

[4] Of course, we are not forgetting the most famous of all, the Linux operating system, that we will confine to a footnote because it is a sort of "bundle" of different OSS with different licensing schemes, rather than a single technology of the type we are dealing with in this paper.

(Leoncini, 2004). Indeed, OSS has questioned the traditional process of software development as a proprietary one. As a matter of fact, recent years have witnessed an increasing competition between proprietary and open source products. In particular, market shares of dominant proprietary vendors have experienced increasing pressure from OSS solutions (see Wheeler (2005) for a thorough review of empirical evidence).

Social science literature started to devote attention to the topic as well.[5]

In this section, we focus on the relationship between OSS and PS, devoting particular attention to the outcomes and implications of the competition between the two software. The literature dealing with the mechanisms of competition and diffusion of the two competing technologies is less developed than the other ones and this is probably due to the difficulty in properly model the process at stake.

The studies in the field can be grouped in two main subsets. The most part of the contributions rely on static models belonging to the industrial organisation literature. In particular, stressing the importance that consumers have in the production process of OSS (Lakhani and von Hippel, 2003), Kuan (2001) models the competition between the two modes of production assuming that agents must decide between buying software and producing it. Johnson (2002) instead models the decision of individual user-programmers to contribute to software program as a decision to contribute to a public good. He shows that programmers participate only if the benefit-cost ratio is higher than a certain threshold and such threshold increases with the probability of free-riding. Schmidt and Schnitzer (2003) identify three main groups of adopters: (i) consumers already using OSS; (ii) consumers already using PS; (iii) users that choose between the two. The authors show that increasing the number of OSS users by means of public provided subsidies can lead to an increase of software price for locked-in proprietary software users. Bitzer (2004) shows that product heterogeneity is the main factor explaining the ability of incumbent firm (in this case, the one adopting a proprietary mode of production) to be profitable by setting a higher price strategy than OSS entering firm. Bessen (2004) constructs a model in which the choice of the form of provision of software is endogenous. Free riding is a less pressing concern, provided that the base product is created in the first place. The main result is that the OSS form of software provision is more efficient because it is able to fulfill more complex and sophisticated needs of consumers.

The second strand of literature is made up of works which try to analyse

[5] In particular, contributions in the field can be organised into five general groups: (i) the investigation of the nature of developers' motivations; (ii) the problem of the governance of OSS projects; (iii) the process of competition between OSS and PS; (iv) the complex topic of IPR influence on OSS and (v) government policies towards OSS. A large part of the literature has mainly concentrated on the first two areas, disregarding almost entirely the others. For a survey, see Rossi (2006).

the issue at stake within a dynamic framework. Their main contribution has been to stress the role played by increasing returns on the demand side. Such works have introduced network effects (both direct and indirect), which are likely to induce path-dependent processes (Arthur, 1989) and to produce lock-in effects (David, 1985). In particular, Bonaccorsi and Rossi (2003) adopt a collective action model and run simulations on a specific explicit OSS adoption function, thus showing that under some plausible assumptions the two software production modes are likely to coexist. Dalle and Jullien (2003) take into account network effects as well, but they distinguish between local and global ones. The former refer to the proportion of a user's neighbours who have already adopted OSS, the latter to the proportion of adopters in the whole population. They run simulations on the OSS adoption function incorporating these factors, together with other more standard ones, and find that the pace of code improvement and proselytism are important factors in explaining OSS success and its coexistence with PS technology. Despite the importance of these contributions, this literature still lacks a tractable analytical treatment of the process, which is very useful if some general results want to be achieved.

Although different in the general framework (static models of industrial organisation vs. non-linear dynamic models), all the models belonging to the two principal subsets reach a similar conclusion: PS and OSS are likely to coexist in the long-run. However, none of these works has been able to analyse properly the dynamics of the diffusion of the two competing software, which are based on different industrial organisation models.[6] To this end, we now revert to models of competition among different technologies. In particular, given the peculiarities of the process at stake, we will refer to the literature on the diffusion of innovation, that we review in the next Section.

2.2 The diffusion of innovation

The literature on the diffusion of innovation is vast and covers different strands, from orthodox to heterodox ones.[7] The literature we refer to in this paper is related to the epidemic models of diffusion, that, starting from the seminal contributions of Mansfield (1961) and Bass (1969), have explored the characteristics of the process of diffusion, explaining it in terms of the disequilibrium process during which knowledge comes to be differentially distributed among agents. In particular, we will follow the strand

[6] A notable exception is the recent contribution of Casadesus-Masanell and Ghemawat (2006), who introduce a dynamic mixed duopoly model allowing competitors to have heterogeneous objective functions and model the presence of demand-side learning. Unlike the standard industrial organisation models, in this paper the dynamics of competition between the two software are properly taken into account.

[7] For a comprehensive survey on models of technology diffusion reporting a set of different models' typologies see, for instance, Karshenas and Stoneman (1995); Geroski (2000).

pioneered by Metcalfe (1981) and Batten (1987), who introduced a supply side in the demand-led model by Mansfield (1961). In this way, the dynamic path follows a logistic pattern determined by the joint dynamics of market demand and growth in production capacity. Unlike the standard diffusion models, where there is no capacity constraint and supply can therefore adjust smoothly to growth in demand, these latter models show how the innovator's reward changes during the diffusion process and this in turn affect supply growth pattern. Further developments brought in the direction of broader models of diffusion (Metcalfe and Gibbons, 1987) have been that of including more than one technique in order to show how the process of diffusion might be the result of competition among techniques, rather than the smooth diffusion of one only (Amable, 1992; Leoncini, 2001).

At the core of the epidemic approach there is the idea that the characteristics of a technology are subject to a progressive path of discovery. The features of the new technology are not well known and, as the available information spreads, the level of uncertainty associated with the new technology decreases, thus increasing the number of adopters.[8] Such models offer a vision of the diffusion process whereby the innovator is rewarded for his/her capacity of supplying a new technology for which the market lacks full information. As the level of uncertainty decreases, expectations play a role, especially if we refer to software markets, as they come to depend on the personal information set an adopter is able to build based on previous adopters' experience, but also on the externalities he/she is able to gather from the network of other users.

These characteristics are at the basis of our choice, as it is fairly evident that software adoption follows a path of progressive discovery of its main features, which is highly dependent on other adopters' information sets. As network externalities are heavily based on information exchange (contagion), this kind of models appears to be the best suited one, although still important modifications have to be made for a correct modelling of the problem. Indeed, the analysis must be enriched with the characteristics peculiar to the software mode of production and to the typology of competition process in action. In particular, two major improvements are worth mentioning: (i) the presence of network effects on the demand side, which is a well known phenomenon discussed in the literature on network industries (e.g. software but also hardware, aircraft, etc.) (Shy, 2001); (ii) the possibility of adopting contemporaneously competing products which is increasingly diffusing in recent years, mainly in industries such as software where technical feasibility of joint adoption has been spurred by increasing rate of standardisation of technologies (Economides, 1996). Thus, the next Section presents a model

[8] Also in more orthodox models, with both one (Karshenas and Stoneman, 1993) and multiple (Stoneman and Kwon, 1994) competing technologies, epidemic effects have been introduced to explain the process of endogenous learning as a process through which information about a technology propagates as that technology spreads in the system.

in line with the tradition of epidemic diffusion models which incorporates all the above mentioned features.

3 General structure of the model

In this Section the general structure of the model is introduced. Before presenting it, some important stylised facts dealing with the task of modelling two competing software technologies on the grounds of epidemic diffusion models should be addressed. In particular, five main points will be considered and properly discussed.

First of all, a common feature characterises the diffusion of software technologies and the diffusion of a disease, namely the fact that both are likely to take place by direct contact. Indeed, most of the time either a new operating system or a new application is likely to be adopted by a non-user if she is persuaded by a current user. Hence, the diffusion of the technology can be thought as a disease that spreads all over a population of non-users who, once infected, add up to the population of current users. This factor points to the adoption of epidemic models to model the process of interest as straightforward.[9]

Second, as pointed out in the previous section, epidemic models take into account the supply side of the story by incorporating the production capacity growth rate (Metcalfe, 1981). This is a reasonable step to be taken if standard technologies are to be modelled. Nonetheless, our analysis concentrate on the software industry and here the relationship between demand and production capacity is not as strict as in the previous case. Specific to the production process characterising software, there is the possibility for the producer to instantaneously supply a new unit of the product at a negligible cost. This fact rules out the need to equate the rate of growth of demand and supply and their level at each point in time. Of course, taking into consideration the supply of software is important and we do it by relating it directly to the speed by which the technology diffuses among users, namely the coefficient of propagation. Indeed, we assume that the two different modes of production characterising the two competing software impact directly on the probability that a user has to infect current non-users.

Third, several contributions in the field have stressed the importance that network effects, both positive and negative, on the demand side have in the software industry (Katz and Shapiro, 1994; Liebowitz and Margolis, 1994; Shy, 2001). While the positive (within-technologies) network effects are already considered in the logistic diffusion process, we model negative

[9] It must be underlined that we will not consider the possibility of differential morbidity among the population of adopters. Indeed, it could be possible to consider that OSS users behave differently from OSS developers. However, as the number of OSS adopters increases, the share of "simple" users increases so that we can assume that OSS and PS users tend to converge as the diffusion process unfolds.

(across-technologies) network effects by means of a parameter, η_i, which has a straightforward interpretation in terms of the lack of interoperability between technology i and technology j. In particular, we conceive η_i as the likelihood that a current user of technology j persuades a user of technology i to dismiss technology i, that is the degree of incompatibility between technology i and technology j.[10] When the degree of incompatibility between PS (x) and OSS (y) increases, then we expect an increase in the likelihood that an OS user will persuade a PS user to dismiss it. Moreover, the degree of interoperability is asymmetric and it depends on the behaviour of software producers.[11] Indeed, the degree of interoperability of OSS with PS – inversely related to η_y – is linked to (i) the willingness of OSS developers to develop interoperable software, and (ii) the degree of closure of PS standards.[12] The degree of interoperability of PS with OSS – inversely related to η_x – is instead mainly due to the decision of PS's producer not to implement OSS standards.

Fourth, switching costs are another important peculiar feature of software industry, strongly affecting the process of competition. Switching costs generally arise when a buyer purchases a particular product repeatedly and he/she finds it costly to switch from one supplier to another. For this reason, in market with switching costs a firm's current market share is an important determinant of its future market success (Klemperer, 1995). We introduce the concept of switching costs by means of the parameter θ_i which should be interpreted as the probability that being a previous user of technology j does not constitute in itself an obstacle to the adoption of i. Thus, the parameter is inversely related with the extent of switching costs between the two technologies which enter into the model in two different ways: (i) the higher θ_i, the higher the potential demand for technology i given the lower costs users of technology j have to face to switch to competing technology; (ii) the higher θ_i, the lower the negative across-technologies network effects.[13]

[10] Although this is not the only kind of interpretation we can attach to parameter η, it is surely the most useful and interesting we can think of for the ease of exposition. Another interpretation can be referred, for example, to the number of available complementary products.

[11] For instance, OpenOffice can read and save files in Microsoft Office formats, whereas the latter does not recognise OpenDocument formats. In a similar manner, the great majority of Linux distributions can read and write file systems in the Microsoft's proprietary formats, while the opposite does not hold, unless by using third party packages.

[12] In case of closed standards, it might still be possible for OSS developers to implement the standard by *reverse engineering*, but this requires more efforts and usually does not produce perfect results.

[13] Provided that our analysis is specific to the software industry, the general concept of switching costs can be better refined in terms of a set of factors peculiar to current developments in the software market. Among the others, the increasing possibility to joint use different software applications and operating systems on the same machine is likely to influence the extent of the switching costs. Regarding PS users, their willingness to

Finally, the two technologies at stake are characterised by two different production processes. Proprietary software is produced by a standard profit-maximising firm, which faces high fixed costs and negligible marginal ones. On the contrary, open source software is produced thanks to the coordination of a community of developers providing the source code for free. Hence, the two supply structures are eminently different. For PS the most important factor influencing its rate of diffusion is the price of the software while, for OSS, the effort lavished by the community of developers turns out to be extremely important.

In line with the previous discussion, we now turn to present the formal structure of the model. First of all, we assume that the maximum number of potential software users (D) is exogenously given. Each software technology diffuses following a logistic pattern. In line with the epidemic approach, every current user of each technology has a given probability to persuade each current non-user to adopt it. The effectiveness of the word-of-mouth ($\beta_i > 0$) can be different for the competing technologies and it is a function of the features of the technology itself.

As for PS (x), we assume that it is a strictly decreasing and concave function of its price ($\beta_x'(p) < 0$, $\beta_x''(p) \geq 0$). Moreover, given that software industry is characterised by economies of scale, both static and dynamic (Shy, 2001), we assume further: $p'(x) < 0$ and $p''(x) > 0$. Hence, the speed of diffusion of PS is a strictly increasing and concave function of its level of adoption:

$$\beta_x'(x) > 0 \tag{1}$$

$$\beta_x''(x) < 0 \tag{2}$$

As for OSS (y), given that its development does not entail any explicit monetary cost, but it is simply the result of the efforts made by the community of developers responding both to intrinsic and extrinsic motivations (Lerner and Tirole, 2002; Bitzer *et al.*, 2007), its final users do not face any direct explicit adoption cost, but only implicit ones.[14] Relying on the fact

joint adopt open source programs is influenced by a number of technical improvements that became popular in recent years. Among the others, CD Linux live distributions that allow users to install and try the operating system without hurting the system already installed are worth mentioning. Moreover, recent commitment of OSS towards user-friendly applications and the development of graphical interfaces similar to existing PS ones has increased the eagerness to joint use PS and OSS once more. On the OSS side, joint adoption of PS programs is fostered via a number of available freeware software which are offered coupled with costly applications or the operating system itself. Furthermore, the marketing strategies – mainly concentrating on the increasing friendliness and on the superior performance and reliability of OSS solutions compared to PS ones – carried out by large proprietary software vendors in recent years goes into this direction as well.

[14]It must be noted that some authors (Wheeler, 2005; Fitzgerald, 2006, e.g.) acknowledge that the total cost of ownership, as implicit cost, may be as high as the explicit costs of buying a PS, although there is not unanimity on this point.

that, the higher the efforts of the community, the more the OSS is "developed" and therefore the less the costs that final users will suffer for using it will be, we assume a negative relation between the total amount of such efforts and the level of these costs. The probability of OSS adoption (β_y) is thus modelled as a strictly increasing function of community's efforts (e): $\beta_y'(e) > 0$, with decreasing marginal returns of efforts, implying: $\beta_y''(e) \leq 0$.

Given that the sets of OSS final users and developers are likely to overlap due to the importance in the OS method of production of user-driven innovation (see Section 2.1 and von Hippel and von Krogh (2003)), we assume that the level of efforts is positively related to that of OSS adoption ($e'(y) > 0$). Moreover, given that larger communities of developers are more likely to face coordination problems, like either disagreement on the actual piece of code to be incorporated into the final release, or disputes over credit attribution, with a higher probability of "forking" (Lerner and Tirole, 2002), we assume further that the increase of efforts in development is less proportional than the increase of the level of adoption ($e''(y) < 0$). Strong evidence supporting this assumption has been recently provided by an empirical work employing data on the population of OSS projects hosted on SourceForge.net (Comino *et al.*, 2007). Hence, the speed of diffusion of OSS is a strictly increasing and concave function of its level of adoption:

$$\beta_y'(y) > 0 \tag{3}$$

$$\beta_y''(y) < 0 \tag{4}$$

Overall, although the production of PS and OSS is driven by different types of actors, firms and community of developers respectively, which respond to different types of incentives, p and e respectively, and, thus, are eminently different in their intimate essence, the hypothesis we made on the relationship between speed of diffusion and level of adoption turns out to be similar for both softwares. This intriguing assumption is based on the main achievements of the literature and, even though we could have refined the functional form of the relationship making it depend on the specific features of each software typology, we decided to keep them as general as possible in order to better grasp the overall implications.

The pattern of diffusion is also affected by negative network effects, linked to the relative level of adoption of the concurrent technology. In particular, we assume that each current user of a technology, given a specific degree of the switching costs, has a constant probability to persuade each user of the other technology to dismiss it. In our case, this probability ($\eta_x \geq 0$) is greater the greater is the lack of interoperability of PS (x) with OSS (y). Factors affecting the extent of across-technologies network effects in the diffusion process might be fruitfully reconnected to the presence of non perfect interoperability between the two technologies.

Finally, as said above, we define a parameter ($0 \leq \theta_i \leq 1$), which proxies the switching costs that each current user of technology j must face to switch to the competing technology i. Let θ_x (θ_y) be the probability that being a previous user of OS (PS) does not constitute in itself an obstacle to the adoption of the competing technology. Indeed, the higher θ_y, the higher the number of agents that OSS users can infect at time t because a lower number of PS users refuse *a priori* to switch to OSS. At the same time, the higher θ_y, the lower the negative network effect exerted by the market share of PS.

The dynamics of diffusion can be thus represented by the following autonomous non-linear system of differential equations:

$$\begin{aligned} \frac{dx}{dt} &= \beta_x(x)\; x\; (D - x - (1-\theta_x)y) - \eta_x(1-\theta_x)\; x\; y \\ \frac{dy}{dt} &= \beta_y(y)\; y\; (D - (1-\theta_y)x - y) - \eta_y(1-\theta_y)\; x\; y \end{aligned} \tag{5}$$

The system reduces to a standard model of Lotka-Volterra equation for two competing species (Hofbauer and Sigmund, 1998) when the following conditions hold: β_i is exogenous, θ_i and η_i go to zero. In this case, β_i is an exogenous propagation coefficient representing the speed of diffusion over the population of non-adopters ($D - x - y$). To sum up, we innovate this baseline model into three main respects: (i) we model the supply of software, being it PS or OSS, by endogenising the propagation coefficient; (ii) we introduce negative network effects on the demand side (positive ones are already considered in the logistic diffusion process); (iii) we take into account the effect of switching costs on both the share of would-be adopters and the extent of network effects.

4 Results

In this section we will present the main results for different specifications of the model. In particular, we will firstly analyse the model by keeping the propagation coefficient constant (Section 4.1). This first specification is discussed in order to show that this particular version of the system resembles the classical Lotka-Volterra one. In Section 4.2 we endogenise the propagation coefficients by making them depend on the price for PS and on developers' effort for OSS respectively. Two cases are then discussed: (i) the simple case with no across-technologies network effects (Section 4.2.1); (ii) the more complex one with such effects (Section 4.2.2).

4.1 Diffusion patterns with a constant propagation coefficient

In this Section, we discuss the results of the model by assuming that the actual level of technology diffusion does not significantly affect the relevant features of the technology, that is, the price for PS and the level of development for OSS ($p'(x) = e'(y) = 0$); or, equivalently, that such features do not alter the probability of adoption ($\beta'_x(p) = \beta'_y(e) = 0$)). Thus, we have that β_i ($i = x, y$) is a constant and the system (5) can be written:

$$\begin{aligned} \frac{dx}{dt} &= \beta_x \; x \left(D - x - (1 + \frac{\eta_x}{\beta_x})(1-\theta_x) \; y \right) \\ \frac{dy}{dt} &= \beta_y \; y \left(D - (1 + \frac{\eta_y}{\beta_y})(1-\theta_y) \; x - y \right) \end{aligned} \tag{6}$$

Equations (6) are the well-known Lotka-Volterra equations for two competing species (see, for instance, Hofbauer and Sigmund, 1998, Ch.3). The isoclines are straight lines with negative slopes:

$$y = \frac{D}{(1+\frac{\eta_x}{\beta_x})(1-\theta_x)} - \frac{1}{(1+\frac{\eta_x}{\beta_x})(1-\theta_x)} x \tag{7}$$

$$y = D - (1 + \frac{\eta_y}{\beta_y})(1-\theta_y) \; x \tag{8}$$

Given the constraints on the parameters, these lines intersect at most once provided that $\frac{\theta_i}{1-\theta_i} \neq \frac{\eta_i}{\beta_i}$ for at least one technology. Figure 1 depicts the possible cases with the out of equilibrium directions.

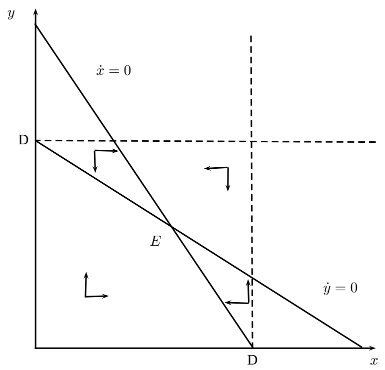
The sufficient and necessary condition for the stable coexistence of the technologies in the market is therefore that the following inequality holds for both technologies:[15]

$$\frac{\theta_i}{1-\theta_i} > \frac{\eta_i}{\beta_i} \tag{9}$$

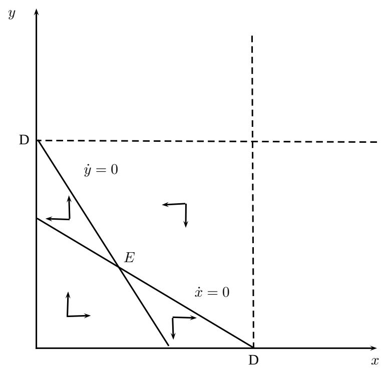
that is, the odds of switching to technology i should be greater than the probability of dismissal for lack of interoperability with j divided by the probability of adoption of i (Figure 1(a)).

If condition (9) does hold for only one technology, this technology displaces completely the other (Figures 1(c) and 1(d)). When instead condition (9) does not hold for any technology, we are in the so called *bistable case* (Figure 1(b)). There are two basins of attraction: the orbits in the first one converge to $(D, 0)$, whereas the others to $(0, D)$, while E is a saddle point. In such case the initial conditions matter.

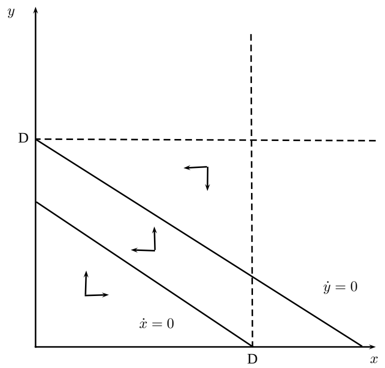
[15] Let us note that this equilibrium is globally stable (or uniformly asymptotically stable in the large). Thus, the initial conditions do not actually matter. For a proof of the global stability of the equilibrium in the case of stable coexistence for the Lotka-Volterra equations for two competing species by means of the Lyapunov function see, for instance, Medio and Lines (2001).



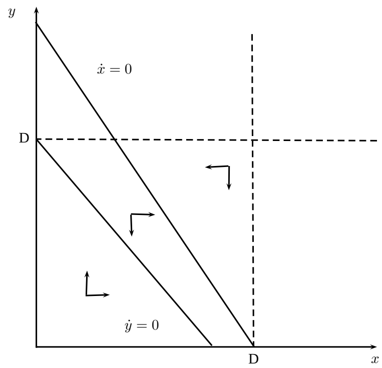
(a) Asymptotically stable equilibrium with technologies' coexistence (high interoperability/low switching costs for both)



(b) Saddle point (low interoperability/high switching costs for both)

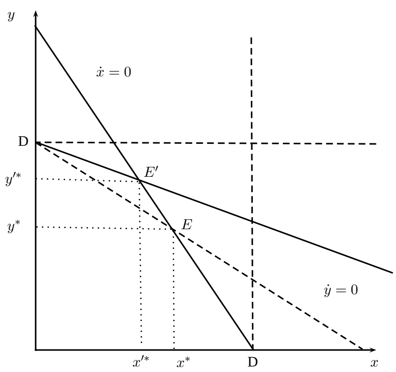


(c) Market tips in favour of OSS (low interoperability of PS/high switching costs for OS users)

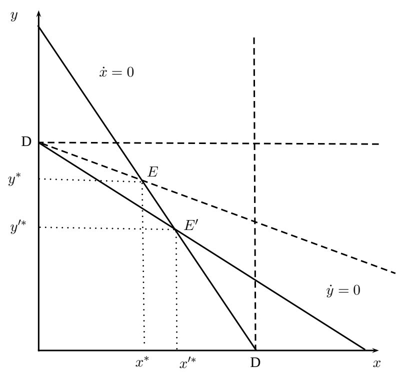


(d) Market tips in favour of PS (low interoperability of OS/high switching costs for PS users)

Figure 1: Dynamics with constant propagation coefficient



(a) Decrease of the switching costs for PS users to adopt OS (θ_y increases)



(b) Closure of PS standards (η_y increases)

Figure 2: Comparative statics

Let us note that, in case of stable coexistence, a decrease of the switching costs to technology i for previous users of technology j decreases (increases) the *total* number of users of technology j (i), both exclusive and co-users; whereas an increase of the interoperability of technology i actually makes these users increase (decrease). In formal terms, if condition (9) is satisfied and the probability of switching is not degenerate ($\theta_i < 1$) for both technologies, we have the following set of partial derivatives:

$$\begin{array}{llll} \frac{\partial x^*}{\partial \theta_x} > 0; & \frac{\partial y^*}{\partial \theta_x} < 0; & \frac{\partial y^*}{\partial \theta_y} > 0; & \frac{\partial x^*}{\partial \theta_y} < 0; \\ \frac{\partial x^*}{\partial \beta_x} > 0; & \frac{\partial y^*}{\partial \beta_x} < 0; & \frac{\partial y^*}{\partial \beta_y} > 0; & \frac{\partial x^*}{\partial \beta_y} < 0; \\ \frac{\partial x^*}{\partial \eta_x} < 0; & \frac{\partial y^*}{\partial \eta_x} > 0; & \frac{\partial y^*}{\partial \eta_y} < 0; & \frac{\partial x^*}{\partial \eta_y} > 0. \end{array}$$

So, for instance, if θ_y increases because of the decrease in the switching costs for previous PS users, there is an absolute increase of OSS users (from y^* to y'^* in Figure 2(a)) and a decrease of PS users (from x^* to x'^* in Figure 2(a)). On the contrary, if the degree of interoperability of OSS technology decreases because of the closure of the standards adopted by PS, this makes total OSS users decrease, whereas PS users instead increases (Figure 2(b)). Thus, the closure of the standard by the PS producer is a very effective strategy in order to tip the market, causing a more than proportional decrease in OSS market share.

When $\beta_i \gg \eta_i$, the number of users who will jointly use the two technologies in equilibrium in case of stable coexistence will be $\left((1 + \frac{\eta_x}{\beta_x})\theta_x - \frac{\eta_x}{\beta_x}\right) y$ $(= \left((1 + \frac{\eta_y}{\beta_y})\theta_y - \frac{\eta_y}{\beta_y}\right) x)$.

Let us finally note that the market shares in case of stable coexistence are not affected by the absolute size of the market.[16]

4.2 Diffusion patterns with a changing propagation coefficient

In this Section we modify the previous model by assuming that the two different supply structures influence the propagation coefficient and, through it, the patterns of diffusion. In particular, Section 4.2.1 assumes perfect interoperability between the two technologies, whereas Section 4.2.2 takes into account the most complex issue of the diffusion of two competing software technologies with non perfect interoperability and switching costs.

[16]The ratio between the total number of OS and PS users is indeed given by

$$\frac{y}{x} = \frac{\beta_x\left(\eta_y(1-\theta_y) - \beta_y\theta_y\right)}{\beta_y\left(\eta_x(1-\theta_x) - \beta_x\theta_x\right)}.$$

and it does not depend on the total amount of demand (D).

4.2.1 Perfect interoperability

To begin with, we analyse the case of perfect interoperability ($\eta_x = \eta_y = 0$). In such case Equations (5) become:

$$\begin{aligned} \frac{dx}{dt} &= \beta_x(x)\ x\ (D - x - (1-\theta_x)y) \\ \frac{dy}{dt} &= \beta_y(y)\ y\ (D - (1-\theta_y)x - y) \end{aligned} \tag{10}$$

Although β_i is now a function of the actual level of diffusion of the correspondent technology, the equilibrium values depend only on the extent of switching costs (θ_i) and are equal to:

$$\begin{aligned} x^* &= \frac{\theta_x D}{\theta_x + \theta_y - \theta_x\theta_y} \\ y^* &= \frac{\theta_y D}{\theta_x + \theta_y - \theta_x\theta_y} \end{aligned} \tag{11}$$

Provided that these parameters are not degenerate, such equilibrium is asymptotically stable, no matter what the actual form of the functions $\beta_i(i)$ is (see Appendix A.1 for a proof of the asymptotic stability of the equilibrium). In such case, the situation is still the one depicted in Figure 1(a), although now there is perfect interoperability. Thus, in the present case the outcome is always the stable coexistence.

It is worth stressing that the market shares of the competing technologies are not affected by those features that interact with the propagation coefficient – the price for PS and level of development for OSS –, but only by the ones which instead affect the switching costs. Thus, by assuming perfect interoperability and the presence of switching costs, the final outcome is neither the most efficient one nor the one in which the product with the best features (effective or potential) is actually chosen (Arthur, 1989). In the stable equilibrium all the users adopt at least one technology, whereas the number of users who jointly adopt the two is equal to $\theta_x y$ ($= \theta_y x$).

4.2.2 Non perfect interoperability

In order to analyse the dynamics in the most complex case, we work out the isoclines:

$$y^*(x) = \frac{1}{1-\theta_x}\frac{\beta_x(x)}{\eta_x + \beta_x(x)}(D - x) \tag{12}$$

$$x^*(y) = \frac{1}{1-\theta_y}\frac{\beta_y(y)}{\eta_y + \beta_y(y)}(D - y) \tag{13}$$

Let us note first that the convex hull of $\{(0,0),(D,0),(D,D),(0,D)\}$ is the only relevant area, given that the threshold D is a physical constraint (i.e. the actual number of users of each technology cannot be greater than the maximum feasible number of users). Hence, we have to take into account only the interval $[0, D]$ for each variable.

To start with, we analyse the isocline of x in such interval. For Equation (12), we have:

$$\frac{dy^*}{dx} = \frac{1}{(1-\theta_x)\left(\eta_x + \beta_x(x)\right)^2}\left(\eta_x \beta'_x(x)(D-x) - \beta_x(x)(\eta_x + \beta_x(x))\right) \quad (14)$$

$$\frac{d^2y^*}{dx^2} = -\frac{2\eta_x(D-x)}{(1-\theta_x)(\eta_x+\beta_x(x))^2}\left(\frac{\beta'_x(x)^2}{\eta_x+\beta_x(x)} + \frac{\beta'_x(x)}{D-x} - \frac{\beta''_x(x)}{2}\right) \quad (15)$$

If conditions (1) and (2) hold and θ_x is not degenerate, we have $d^2y^*/dx^2 < 0$ for $x \in [0, D]$. Hence, Equation (12) is strictly concave in such interval. Moreover, given that:

$$\left.\frac{dy^*}{dx}\right|_{x=D} = -\frac{1}{1-\theta_x}\frac{\beta_x(D)}{\eta_x+\beta_x(D)} \ (<0)$$

and $\lim_{x\to D} y^*(x) = 0$, by the strict concavity of $y^*(x)$ follows that:

$$y^*(x) = y^*(D + (x-D)) < \frac{1}{1-\theta_x}\frac{\beta_x(D)}{\eta_x+\beta_x(D)}(D-x)$$

for each $x \in [0, D)$. Thus, the function $y^*(x)$ lies below the straight line:

$$y = \frac{D}{(1+\frac{\eta_x}{\beta_x^M})(1-\theta_x)} - \frac{1}{(1+\frac{\eta_x}{\beta_x^M})(1-\theta_x)}x \quad (16)$$

where $\beta_x^M = \beta_x(D)$. This line is the isocline of x in the model with a constant propagation coefficient, calculated at the maximum attainable PS propagation coefficient (Equation (7)).

Moreover, by assuming $\lim_{x\to 0}\beta_x(x) = 0$, we have:

$$\lim_{x\to 0}\frac{dy^*}{dx} = \frac{1}{1-\theta_x}\frac{\beta'_x(0)}{\eta_x}D > 0$$

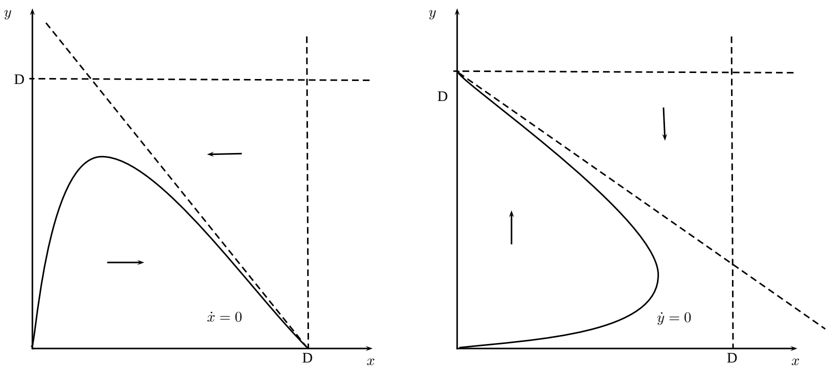
and there is therefore a unique local maximum of the function ($y_M^* \in [0, \frac{D}{1-\theta_x}]$) lying in the domain $(0, D)$.[17] Hence, the function is as in Figure 3(a) and x will increase (decrease) depending on the combination (x, y) being actually below (above) the function.

[17] When $\eta_x = 0$ we have:

$$\lim_{x\to 0} y^*(x) = \lim_{x\to 0}\frac{\beta_x(x)D}{(1-\theta_x)\beta_x(x)} = \lim_{x\to 0}\frac{\beta'_x(x)D}{(1-\theta_x)\beta'_x(x)} = \frac{D}{(1-\theta_x)}.$$

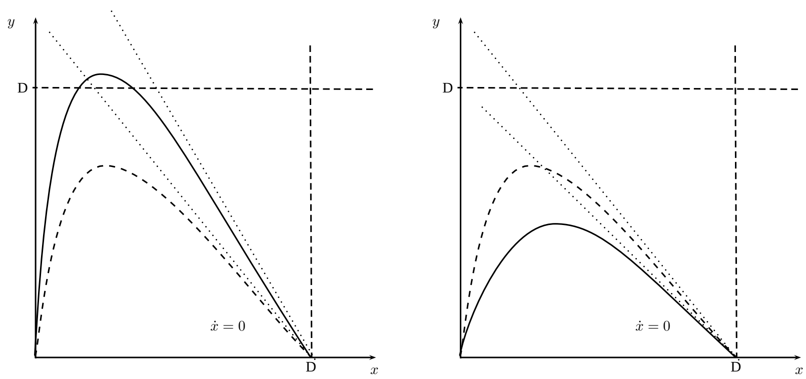
$$\lim_{x\to 0}\frac{dy^*}{dx} = \lim_{x\to 0} -\frac{\beta_x(x)^2}{(1-\theta_x)\beta_x(x)^2} = \lim_{x\to 0} -\frac{\beta'_x(x)\left(\beta'_x(x)+\beta''_x(x)\right)}{(1-\theta_x)\beta'_x(x)\left(\beta'_x(x)+\beta''_x(x)\right)} = -\frac{1}{1-\theta_x}$$

and we are back in the case analysed in Section 4.2.1.



(a) Isocline of x (b) Isocline of y

Figure 3: Isoclines



(a) Increase of θ_x (b) Increase of η_x

Figure 4: Changes of the parameters

What remains to be analysed are the effects of the two parameters (θ_x and η_x) on the shape of the function. As for θ_x, it is sufficient to note that it enters the function simply as a multiplicative constant. Hence, an increase of θ_x moves x isocline upward as in Figure 4(a), whereas a decrease of it moves the curve the other way around. As for η_x, an increase of it makes the curve change as in Figure 4(b) (see Appendix B for a more in depth analysis of the effects of the parameters on $y^*(x)$).

In the light of the analogy of assumptions (1)-(2) and (3)-(4), Equations (12) and (13) are symmetric with respect to the axes and the isocline of y is the one shown in Figure 3(b). All the analysis carried out for the isocline of x is therefore valid also for the other isocline provided that the notation for x is substituted with the notation for y.

The solutions for the most complex case are depicted in Figure 5. Again, depending on the shape of the isoclines, we can have one stable solution with coexistence of both technologies (point E in Figure 5(a)).[18] However, E is not a globally stable equilibrium, as points outside the hearth-shaped area tend towards equilibria characterised by winner-take-all solutions (points D). This is even more the case if the isoclines move, making the area of stable trajectories to shrink, until the equilibrium point E becomes a saddle point, which shows one winning technology and dependence from initial conditions (Figure 5(b)).

Once comparative dynamics are taken into consideration, some interesting issues emerge (Figure 6). First of all, a closure of the PS standard is likely to induce a strong negative effect on the OS technology which can even tip the market in favour of PS (Figure 6(a)). This highlights the fact that PS producer has an important strategic instrument to reduce the degree of interoperability of OSS and lock-in the market towards proprietary technology. Second, a decrease in the level of switching costs for PS users produces an increase in the overall number of users adopting OSS (Figure 6(b)). This increase can be a consequence of the commitment of OSS towards user-friendly applications and the development of graphical interfaces similar to existing PS ones.

It should also be noted that, as far as the OS technology is concerned, the community ethos, i.e. the strong sense of belonging to the community of developers, has instead a strong negative effect on θ_x, because it decreases considerably the probability of OS users to adopt the PS technology. [19]

Finally, it might happen that, with a low level of interoperability for both technologies, no diffusion actually takes place (Figure 6(c)). Nevertheless, in this case the more likely outcome is that the technology characterised by even a little advantage locks the market in.

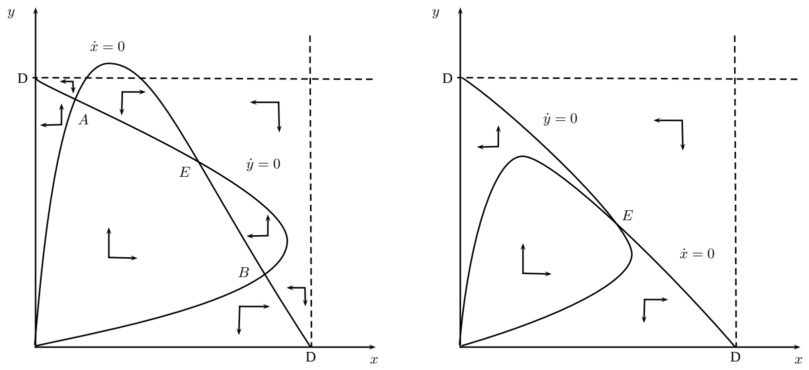
5 Conclusions

This paper has shown how to implement a formal model of innovation diffusion to model two competing technologies in high-tech industries. Indeed, as they are characterised by the presence of economies of scale, within- and across-technologies network effects and switching costs, a proper theoretical modeling is needed, that, to our knowledge, has not been carried out so far. Moreover, if open source is taken into consideration, other issues surface, such as the effort of the community and developers' motivations.

All these topics have been incorporated in a modified version of a standard epidemic model, which innovate the existing literature in many respects. Above all, the endogenisation of the propagation coefficient adds

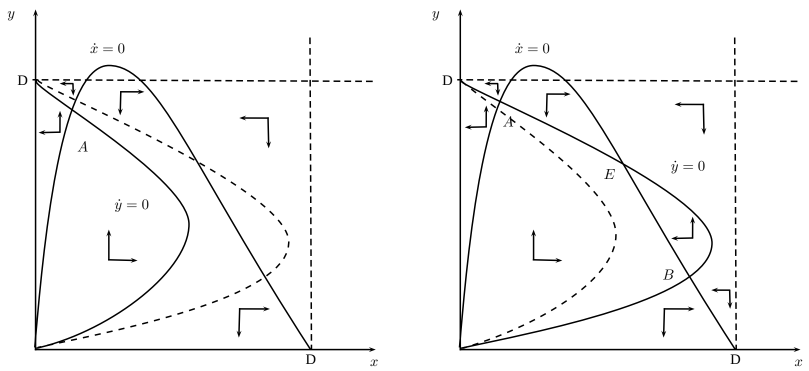
[18]See Appendix A.2 for a proof of the stability of the equilibrium.

[19]In this sense, for example, the diffusion of the Free Software movement and the creation of Free Software Foundation (1984), along with the strong charisma of his founder, Richard Stallman, has been a great tool of diffusion.

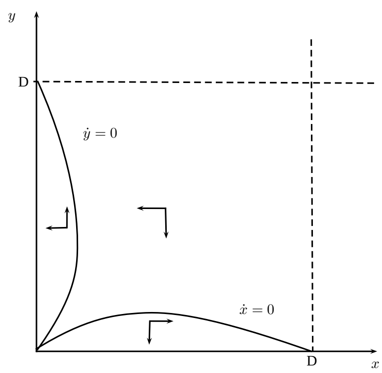


(a) Stable node (b) Saddle point

Figure 5: Fixed points



(a) Decrease of OSS interoperability (b) Decrease of switching costs for PS users



(c) Strong lack of interoperability

Figure 6: Comparative dynamics

complexity to the general structure of the model, yielding interesting results, such as the coexistence of an asymptotically stable equilibrium where both technologies survive with winner-take-all solutions. This result is obtained in a dynamic setting, thus enriching the achievements of the literature on OSS-PS competition, which obtained it only in a static context. Moreover, our model adds up new insights to the literature on the diffusion of competing technologies under increasing returns where the standard result is that the market tips in favour of one of the two. Indeed, the process of competition between technologies has been modeled by the literature as a situation where either one technology tips the market for an indefinite period of time (David, 1985; Arthur, 1989; Amable, 1992) or a superior technology, after being adopted by a critical mass of users, displaces the other one (Witt, 1997; Andreozzi, 2004). On the contrary, we show how both coexistence and market tipping are likely outcomes also in a situation where both technologies start the competition process at the same time. Finally, network effects and switching costs turn out to be important factors that the supplier of one technology can change in order to alter the equilibrium point and thus its market share.

Within this innovative theoretical framework, the main results are the following. First of all, in all the different specifications of the model there is always the possibility to obtain an asymptotically stable equilibrium where both technologies coexist. Thus, contrary to the result obtained by Amable (1992), the process of competition between two technologies characterised by increasing returns do not necessarily ends up with one of the two tipping the market, at least in the present case, where the possibility of joint adoption is taken into account.

Second, we have been able to present the conditions that determine the success of one technology with respect to another one: (i) under the assumption of a constant propagation coefficient (with the condition given by $\frac{\theta_i}{1-\theta_i} > \frac{\eta_i}{\beta_i}$), the odds of switching to competing technology must be greater than the ratio of the probability of dismissal for lack of interoperability and the (constant) probability of adoption; (ii) when the coefficient of propagation is let to vary, then the condition can be computed formally only if a functional form for the propagation coefficient is assumed.

Third, in the case of coexistence of both technologies, it is possible to modify the equilibrium by changing the values of parameters θ and η. In particular, a decrease of the switching costs for PS users (an increase of θ_y) yields a more than proportional increase in the number of OSS users and a contemporary decrease in the absolute number of PS users. If the negative network effect increases (an increase of η_y), then the number of OSS users decreases and PS users increase.[20]

[20]These are mainly comparative statics' exercises and not a complete strategic analysis which constitutes food for thought of a future paper.

Finally, the stability of the equilibrium point is a recursive result through the different specifications of the model. However, while under the assumption of exogenous probability of adoption, the equilibrium point is globally stable, when the propagation coefficient is endogenous the equilibrium point is only locally stable and it coexists with two other points around which one of the two technologies is likely to tip the market. In this last case, a change in the parameters θ and η leads to a modification of the basin of attraction of the equilibrium point.

References

Amable, B. (1992) Competition among techniques in the presence of increasing returns to scale, *Journal of Evolutionary Economics*, 2, pp. 147–158.

Andreozzi, L. (2004) A note on critical masses, network externalities and converters, *International Journal of Industrial Organization*, 22(5), pp. 647–653.

Arthur, B. (1989) Competiting technologies, increasing returns, and lock-in by historical events, *The Economic Journal*, 99(394), pp. 116–131.

Bass, F. M. (1969) A new product growth for model consumer durables, *Management Science*, 15, pp. 215–227.

Batten, D. (1987) The balanced path of economic development: a fable for growth merchants, in: D. Batten, J. Casti and B. Johansson (Eds.) *Economic Evolution and Structural Adjustment* (Springer-Verlag).

Bessen, J. (2004) *Open Source Software: Free Provision of Complex Public Goods*, Working paper, SSRN.

Bitzer, J. (2004) Commercial versus open source software: the role of product heterogeneity in competition, *Economic Systems*, 28(4), pp. 369–381.

Bitzer, J., Schrettl, W. and Schroeder, P. J. (2007) Intrinsic motivation in open source software development, *Journal of Comparative Economics*, 35(1), pp. 160–169.

Bonaccorsi, A. and Rossi, C. (2003) Why open source software can succeed, *Research Policy*, 32(7), pp. 1243–58.

Brynjolfsson, E., Kemerer, C. and of Management, S. S. (1996) Network Externalities in Microcomputer Software: An Econometric Analysis of the Spreadsheet Market, *Management Science*, 42, pp. 1627–1647.

Casadesus-Masanell, R. and Ghemawat, P. (2006) Dynamic mixed duopoly: A model motivated by linux vs. windows, *Management Science*, 52(7), pp. 1072–1084.

Chiaravutthi, Y. (2006) Firms'strategies and network externalities: Empirical evidence from the browser war, *Journal of High Technology Management Research*, 17(1), pp. 27–42.

Comino, S., Manenti, F. and Parisi, M. (2007) From planning to mature: On the success of open source projects, *Research Policy*, 36(10), pp. 1575–1586.

Dalle, J. M. and Jullien, N. (2003) 'libre' software: Turning fads into institutions?, *Research Policy*, 32(1), pp. 1–11.

David, P. (1985) Clio and the economics of QWERTY, *American Economic Review: Papers and Proceedings*, 75, pp. 332–337.

Economides, N. (1996) The economics of networks, *International Journal of Industrial Organization*, 14(6), pp. 673–699.

Fitzgerald, B. (2006) The Transformation of Open Source Software, *Management Information Systems Quarterly*, 30(3), p. 587.

Fosfuri, A., Giarratana, M. and Luzzi, A. (2008) The penguin has entered the building: The commercialization of open source software products, *Organization Science*, 19(2), p. 292.

Gandal, N. (1994) Hedonic price indexes for spreadsheets and an empirical test for network externalities, *RAND Journal of Economics*, 25(1), pp. 160–170.

Gandal, N. (1995) Competing compatibility standards and network externalities in the pc software market, *The Review of Economics and Statistics*, 77(4), pp. 599–608.

Geroski, P. A. (2000) Models of technology diffusion, *Research Policy*, 29(4-5), pp. 603–625.

Hofbauer, J. and Sigmund, K. (1998) *Evolutionary Games And Population Dynamics* (Cambridge: Cambridge University Press).

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

Karshenas, M. and Stoneman, P. (1993) Rank, stock, order, and epidemic effects in the diffusion of new process technologies: An empirical model., *RAND Journal of Economics*, 24(4), pp. 503–28.

Karshenas, M. and Stoneman, P. (1995) Technological diffusion, in: P. Stoneman (Ed.) *Handbook of the Economics of Innovation and Technical Change* (Basil Blackwell).

Katz, M. L. and Shapiro, C. (1994) Systems competition and network effects, *Journal of Economic Perspectives*, 8(2), pp. 93–115.

Klemperer, P. (1995) Competition when consumers have switching costs: An overview with applications to industrial organization, macroeconomics, and international trade, *The Review of Economic Studies*, 62(4), pp. 515–539.

Kuan, J. W. (2001) *Open Source Software as Consumer Integration Into Production*, Working paper, SSRN.

Lakhani, K. R. and von Hippel, E. (2003) How open source software works: Free user-to-user assistance, *Research Policy*, 32(6), pp. 923–943.

Leoncini, R. (2001) Segmentation and increasing returns in the evolutionary dynamics of competing techniques., *Metroeconomica*, 52(2), pp. 217–237.

Leoncini, R. (2004) Of penguins and innovation, *Istituzioni e Sviluppo Economico*, 3(2), pp. 57–71.

Lerner, J. and Tirole, J. (2002) Some simple economics of open source, *Journal of Industrial Economics*, (52), pp. 197–234.

Liebowitz, S. J. and Margolis, S. E. (1994) Network externality: An uncommon tragedy, *Journal of Economic Perspectives*, 8(2), pp. 133–150.

Mansfield, E. (1961) Technical change and the rate of imitation, *Econometrica*, 29, pp. 741–766.

Medio, A. and Lines, M. (2001) *Non linear dynamics. A primer* (Cambridge: Cambridge University Press).

Metcalfe, J. (1981) Impulse and diffusion in the process of technological change, *Futures*, pp. 347–359.

Metcalfe, J. and Gibbons, M. (1987) On the economics of structural change and the evolution of technology., in: L. Pasinetti and P. Lloyd (Eds.) *Structural Change, Economic Interdependence and World Development* (St. Martin's Press).

Rossi, M. A. (2006) Decoding the "free/open source(f/oss) software puzzle" a survey of theoretical and empirical contributions, in: J. Bitzer and P. Schroder (Eds.) *The Economics of Open Source Development* (Elsevier).

Schmidt, K. M. and Schnitzer, M. (2003) *Public Subsidies for Open Source? Some Economic Policy Issues of the Software Market*, CEPR Discussion Papers 3793.

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

Stoneman, P. and Kwon, M. J. (1994) The diffusion of multiple process technologies., *Economic Journal*, 104(423), pp. 420–431.

von Hippel, E. and von Krogh, G. (2003) Open source software and the private-collective innovation model: Issues for organization science, *Organization Science*, 14(2), pp. 209–223.

West, J. (2003) How open is open enough? melding proprietary and open source platform strategies, *Research Policy*, 32(7), pp. 1259–1285.

Wheeler, D. (2005) Why open source software / free software (oss/fs, floss, or foss)? look at the numbers!, available at http://www.dwheeler.com/oss_fs_why.html .

Witt, U. (1997) "lock-in" vs. "critical masses" – industrial change under network externalities, *International Journal of Industrial Organization*, 15(6), pp. 753–773.

Appendix

A Proof of the asymptotic stability of the equilibrium

A.1 Perfect interoperability

In order to prove the stability of the equilibrium in the case of perfect interoperability analysed in Section 4.2.1, let us work out the Jacobian of the system at the equilibrium:

$$\mathbf{J} = \begin{bmatrix} -\beta_x(x^*)x^* & -(1-\theta_x)\beta_x(x^*)x^* \\ -(1-\theta_y)\beta_y(y^*)y^* & -\beta_y(y^*)y^* \end{bmatrix}$$

The discriminant of the associated characteristic equation is:

$$\begin{aligned} \Delta &= (-\beta_x^* x^* - \beta_y^* y^*)^2 - 4\left((1-(1-\theta_y)(1-\theta_x))\,\beta_x^* x^* \beta_y^* y^*\right) = \\ &= (\beta_x^* x^* - \beta_y^* y^*)^2 + 4(1-\theta_y)(1-\theta_x)\beta_x^* x^* \beta_y^* y^* > 0 \end{aligned}$$

The determinant of $\mathbf{J}$ is positive whereas its trace is negative, therefore both the eigenvalues are real and negative and (x^*, y^*) is a stable node.

A.2 Non perfect interoperability

In the more general case of non perfect interoperability (Section 4.2.2), the Jacobian calculated at the fixed points is:

$$\mathbf{J} = \begin{bmatrix} -\beta_x(x^*)x^* & -(\eta_x + \beta_x(x^*))(1-\theta_x)x^* \\ -(\eta_y + \beta_y(y^*))(1-\theta_y)y^* & -\beta_y(y^*)y^* \end{bmatrix}$$

The discriminant of the associated characteristic equation is thus:

$$\begin{aligned} \Delta &= (\beta_x^* x^* + \beta_y^* y^*)^2 - 4\left(\beta_x^* x^* \beta_y^* y^* - (\eta_x + \beta_x^*)(\eta_y + \beta_y^*)(1-\theta_y)(1-\theta_x)x^* y^*\right) = \\ &= (\beta_x^* x^* - \beta_y^* y^*)^2 + 4(\eta_x + \beta_x^*)(\eta_y + \beta_y^*)(1-\theta_y)(1-\theta_x)x^* y^* (> 0) \end{aligned}$$

This discriminant is always positive, whereas the trace of the Jacobian is negative. Thus, the fixed points can be either saddle points or stable nodes depending on the determinant of the Jacobian being positive or negative. This determinant is equal to:

$$|\mathbf{J}| = \beta_x^* x^* \beta_y^* y^* - (\eta_x + \beta_x^*)(\eta_y + \beta_y^*)(1-\theta_y)(1-\theta_x)x^* y^*$$

and it is positive if and only if:

$$(1-\theta_y)(1-\theta_x)(1 + \frac{\eta_x}{\beta_x^*} + \frac{\eta_y}{\beta_y^*} + \frac{\eta_x}{\beta_x^*}\frac{\eta_y}{\beta_y^*}) < 1$$

that is, if:

$$-\frac{1}{(1-\theta_x)(1+\frac{\eta_x}{\beta_x^*})} < -(1-\theta_y)(1+\frac{\eta_y}{\beta_y^*}) \tag{17}$$

Inequality (17) is satisfied in point E of Figure 5(a). Indeed, in such point we have:

$$\frac{dy^*}{dx} < \frac{1}{\frac{dx^*}{dy}}$$

Given that, from Equation (14) it follows that:

$$\frac{dy^*}{dx} > -\frac{1}{(1-\theta_x)(1+\frac{\eta_x}{\beta_x^*})}$$

Recalling the symmetry between Equation (12) and (13), we have:

$$-\frac{1}{(1-\theta_x)(1+\frac{\eta_x}{\beta_x^*})} < \frac{dy^*}{dx} < \frac{1}{\frac{dx^*}{dy}} < -(1-\theta_y)(1+\frac{\eta_y}{\beta_y^*})$$

and the point E is therefore a stable node.

B Effects of parameter changes on the isocline

As for θ_x, by the envelope theorem the marginal effect of an increase of it on the maximum of $y^*(x)$ (y_M^*) is:

$$\frac{\partial y_M^*}{\partial \theta_x} = \frac{y_M^*}{1-\theta_x}(>0)$$

This marginal effect is therefore directly proportional to the initial level of the maximum and it increases for increasing values of the parameter ($\frac{\partial^2 y_M^*}{\partial \theta_x^2} > 0$).

As for η_x, its marginal effect is:

$$\frac{\partial y_M^*}{\partial \eta_x} = -\frac{y_M^*}{\eta_x + \beta_x(y^{*-1}(y_M^*))}(<0)$$

Also this effect is directly proportional to the initial level of the maximum and it increases for increasing values of the parameter ($\frac{\partial^2 y_M^*}{\partial \eta_x^2} < 0$).

Moreover, a change of η_x makes also the value of x corresponding to y_M^* change. In particular, an increase of η_x makes $y^{*-1}(y_M^*)$ decrease if and only if:

$$\beta_x(y^{*-1}(y_M^*)) > \beta_x'(y^{*-1}(y_M^*))(D - y^{*-1}(y_M^*)). \tag{18}$$

Indeed, from Equation (14) it follows that the FOC are satisfied if the expression in brackets is equal to zero. Working out the total differential of such expression and equating it to zero, after some algebraic manipulation we obtain:

$$\frac{dx}{d\eta_x} = \frac{\beta_x(x) - \beta_x'(x)(D - x)}{\beta_x''(x)\eta_x(D - x) - 2\beta_x'(x)(\eta_x + \beta_x(x))}.$$

The denominator of such expression is negative. Therefore it is negative if and only if condition (18) holds.