Open Source and Commercial Software Platforms: Is Coexistence A Temporary or Sustainable Outcome?

Eric Darmon, Université de Rennes 1 and CREM-CNRS, France
Dominique Torre, GREDEG-CNRS and University of Nice - Sophia Antipolis, France

ABSTRACT

In this article, we consider the dynamics and competition between two software platforms (Open Source and proprietary software). Potential user-developers can adopt one of the two platforms in order to develop and sell new applications based on the platform. We consider the static issue first and then use a simple dynamic system where the dynamics comes from the development efforts (spillovers) made on each platforms. In this context, we first identify the conditions for the two platforms to coexist in the long run. From this baseline, we then consider different strategies for the editor. A first strategy is for the editor to “show the code” of its software, so as to develop more compatible products. A second strategy is to strategically monitor the compatibility degree between the proprietary and OS platform. In both cases, we analyse whether a mixed industry may be sustainable in the long run. [Article copies are available for purchase from InfoSci-on-Demand.com]

Keywords: Compatibility Strategy; Open Source; Platform; Software Development; Software Industry

INTRODUCTION

Open Source (OS) Software have attracted much attention in the last decade. From an economic point of view, the initial focus was on the original production and distribution modes of Open Source Software (see e.g. Bitzer and Schröder, 2006 for a survey) and the public good properties and their direct impacts – developers’ motivations to engage in OS communities (see e.g. Lerner & Tirole, 2002) were stressed. As a second step, the original competition between an OS (non profit motivated) organization and a proprietary/commercial (profit motivated) software editor has been studied (see e.g. Bonnacorsi and Rossi [2002], Darmon, Le Texier and Torre [2007], see also Gaudeul and Pollock in this issue). Now, OSS seems to impose in many sectors so that interactions between the commercial and OS software production mode
have taken new forms and both actors need to reconsider their strategies.

On the one hand, OS seems to have obtained large market shares in many different areas so that incumbent software editors need now to change their attitude towards OS. Looking at Microsoft’s strategy as some anecdotic evidence, one can observe how Microsoft’s strategy towards OS has tremendously evolved from a competition strategy to a somewhat more cooperative approach (see e.g. Galoppini [2008]). Conversely, while many firms now think of “going open source”, many OS projects consider the reverse way and now think of going “business”. These two simultaneous trends lead to new types of “hybrid models” within the software industry. Such models may take very diverse forms. Dual licences can also be considered as a first hybrid strategy where a firm chooses to distribute two differentiated products: one is released under proprietary licence terms while the other is OS (see e.g. Comino and Manenti [2008] for an economic analysis). Another strategy goes through a better interoperability between each OS and Proprietary platforms or software. Many OS developers are firms that are users-developers à la Von Hippel [2005]. They decide to use OS primarily because of the higher utility they can derive from it. Besides, it should be kept in mind that many open-source developers are directly employed by firms to contribute to OS communities (see Henkel [2006 a, b] for an illustration to embedded Linux). Considering the case of a platform software (i.e. a software that other firms/users adopt to build new pieces of software under that platform), the commercial editor of such software platform may think of “opening” its code in order to increase the utility of its users and increase its market share when confronted to an OS competitor.

In a framework close to this article, Economides and Katsamakas [2007] compare a situation where the software platform is proprietary to another where it is OS. They show the OS outcome may be socially more efficient depending on users’ preference for application variety. Parker and Van Alstyn [2008] investigate a close situation where the firm can maximize the development and network externalities when it selects an open source-based strategy. These papers yield interesting results as regards with welfare and firms strategy. Yet, they do not explicitly consider the rivalry between two competing platforms and the long term effects of such competition.

As Casadesus-Masanell and Ghemawat [2006], we consider in this article a situation where proprietary and OS platforms are simultaneously active on the market. While their paper focuses much on demand-side externalities, we here focus on knowledge externalities. In our model, there are two platforms that can then be adopted by independent developers and by end users to build new pieces of software and to introduce new functionalities. When confronted to an OS competitor, the proprietary platform may have an incentive to go open. It can first “show the code” to its users. This would enable users to build more compatible applications thus leading to a higher utility of the proprietary platform. Second, the editor can decide to rearrange its compatibility strategically so that it may capture larger spillovers. In both cases, we study the incentive for a software editor to implement such strategy in the short term. Using a simple dynamical system, we determine in each case, the set of conditions for which such competition would lead to a sustainable hybrid industry in the long run.

THE BENCHMARK MODEL

There are four categories of agents. The two first agents are the editor (or publisher) of the proprietary software and the Open Source community/project. These two agents develop two competing software platforms. Rights to use this platform are defined according to the licence terms of the two software. OS licence terms are here General Public Licence (GPL). In the basic model, the publisher uses a standard proprietary model (closed source code, copyright product). The two other agents are “final” users and “developers” respectively. Users have
Open Source E-Learning Systems: Evaluation of Features and Functionality
www.igi-global.com/article/open-source-e-learning-systems/101216?camid=4v1a