Modeling production in the Creative Commons

Massimiliano Gambardella[12] *& Matthijs den Besten*[3]

Abstract

Creative Commons (CC) licenses are increasingly used and the number of works under these licenses is growing. However, for each successful project there are many others that fail because they are unable to attract user contributions. Soliciting the contributions of users is a challenge for the management of a CC project. The aim of this paper is to shed light on the factors that contribute to the success of a CC project. To do that we develop an agent-based model that simulates the hidden dynamics of the production of CC works. This model is able to replicate stylized facts of CC production. Moreover, the model shows that characteristics of the CC project, such as the effort necessary to complete the project, the prestige of the producer, and its legal status are fundamental to its success.

1 Introduction

The number of projects under Creative Commons (CC) licenses has increased rapidly over the last ten years, piquing our desire to understand the factors involved in the production of works under these licenses.

[1] Legislative Office of Groups, Chamber of Deputies (Italy)
[2] University Paris Ouest Nanterre – La Défense
[3] Groupe Sup de Co Montpellier Business School, Montpellier Research in Management

The CC are a set of licenses directly derived from Free *Libre* Open Source Software (FLOSS) licenses. Unlike FLOSS, CC are used in art productions. Like FLOSS, the aim of CC licenses is to facilitate the sharing of works among people and the collection of contributions from users (Lessig 2004).

Despite the extensive work on FLOSS production, there is little literature analyzing CC production (Mustonen 2010).

The collection of data is more difficult for projects under CC than under FLOSS licenses. In the case of FLOSS all information on the production process and the contributions of users is stored in online platforms (such as SourceForge, GitHub, Bitbucket, Google Code, etc.) used by developers. Conversely, in case of videos, music, or texts under CC licenses only the final result of the production process is stored on online platforms (such as YouTube, Internet Archive, DailyMotion, Flicker, Picasa, Spotify, SoundCloud, etc.).

For each successful CC project there are many others that never succeed, failing to attract contributions from users. Soliciting the contributions of users is a challenge for the management of a CC project. The aim of this paper is to understand the paths (David 1985) and factors that contribute to the success of a CC project.

Motivational rules and objective functions govern the behavior of producers and users who contribute to a work. This behavior can be described by a model in which is possible to observe consistent "emergent properties" (Dalle and David 2005). To do that, we developed the basic structure of a model that reproduces the stylized facts observable in real life (Malerba, Nelson et al. 2006). This model is an Agent-Based Model (ABM) (Tesfatsion and Judd 2006) that reproduces the stylized facts of a production of artistic works under CC licenses. Specifically, in this model the output is a selection among video projects under CC licenses, represented by the stylized facts. As mentioned above, this output can be empirically observed on online platforms. For purposes of this study we decided to replicate empirical data (the output of the video-project selection process) extracted from the Internet Archive (IA). This output was used to develop and calibrate an ABM that can simulate the hidden patterns present in CC production. We decided to use the IA because it is the platform recommended by the Creative Commons Organization for storing CC works.[4]

On IA we observe different creators, such as video-makers, dentists, software houses, the U.S. Congress, Nonprofits and many others having different goals, business models, and strategies, and which create videos for a variety of purposes. Video-makers do it to disseminate their films, dentists to promote their businesses, software houses to give video guides to their users, etc. For purposes of this study we do not differentiate between the various objectives of different creators.

Each work starts from a project. If the project succeeds the work is done. A project succeeds if it is able to collect a sufficient number of contributions. Different types of contributions might be required to complete a project: funds, unpaid work, feedback, etc. In this study we do not distinguish between the different types of contributions; a contribution is considered a single unit of something that is necessary to complete the project. When the project is completed we have the final work. Then the work is stored and observable on the online platform.

[4] http://creativecommons.org/weblog/entry/7629

However for each project that succeeds there are many others that fail. Our agent-based model is able to simulate the dynamics of such selection process.

We assume that a project succeeds when it is completed. Thus, the success of a CC project is attributable to its capacity to attract a sufficient number of contributions.

Participants in CC production are heterogeneous and their decisions to allocate efforts reflect different motivations. The ABM is an excellent tool to focus on individuals with heterogeneous behavior (Radtke, Janssen et al. 2009).

The model universe consists of agents and projects. Agents may choose to create a new project, to contribute to an existing one, or to do nothing. We assume that the creator of a new project acts as producer and that contributors act as users.

Our ABM consists of a *prior-platform* that is a sort of virtual market in which agents propose their projects, decide which one to contribute to, or opt to do nothing. Agents decide to participate in a project on the basis of their utility functions. Each utility function accounts for different factors reflecting the characteristics of each project. The output of this virtual market is the list of completed projects that we can empirically observe on the online platform (the IA).

This model can be used to explore the conditions necessary for eliciting the contributions of users, who play a fundamental role in determining the success of CC projects.

This paper contains 5 sections. In the next section we present the influential factors of the model and a description of the model itself. In the following section we describe the calibration and validation methods. Then we introduce the results of our model and formulate an econometric model to run on the output of the ABM. Finally we discuss the results of both the output of the model and the regression analysis.

2 The Model

The model is a tool to help generate hypotheses about what process might have generated the results observed on IA. The model itself defines the structure of the process. In our case, participants with given preferences are selected randomly to initiate new projects or to contribute to existing projects or to do nothing.

Our hypotheses are on the preferences and characteristics of the participants. In order to explore a range of assumptions that might lead to the desired outcome, we run simulations of the model with different instantiations of these preferences and characteristics. Then we observe the average outcome of each simulation to check how close it is to our benchmark, the empirical data.

Our benchmark spans several periods. For each period we can imagine the likely preferences and characteristics of participants based on our intuition and corroborated by the model output. Based on these exercises for each period, we can then reconstruct the likely evolution of preferences and characteristics in CC production.

3.1 Influential Factors

A fundamental step in developing an ABM is to identify the main factors that are most likely to influence the dynamics. We identify two categories of influential factors:

1. The characteristics of the projects
2. The motivations of the users

The *characteristics of the projects* are features directly related to projects and their production. Characteristics can be subjective (beautiful or ugly, interesting or boring, etc.) or objective (completed or not, started by a Nonprofit producer or not, started by a prestigious producer or not, the number of contributions necessary for it to be completed). For these purposes we create variables corresponding to the objective characteristics of the projects:

"Success" is the dependent variable that represents the success of a project—the work is done and published on the online platform. It is the production average of a single producer. This average is calculated from the different scenarios of the different years.

"Commercial" is an independent variable. It is randomly assigned to the model. It is a dummy variable that indicates whether a For-Profit or a Nonprofit producer started the project. To be produced by a For-Profit is considered to have a negative impact on the success of a project (Valentinov 2007) (Gambardella 2011).

"Prestigious" is an independent variable. It is randomly assigned to the model. It is a dummy variable that indicates whether a prestigious producer started the project. To be started by a prestigious producer is considered conducive to the success of the project. We assume that prestige helps to promote the project, attracting users motivated by *reputation*, *peer-recognition* and *career concerns* (Lerner and Tirole 2005).

"Maturity" is an independent variable. It is randomly assigned to the model. This variable indicates the remaining level of effort required to complete the project. To require greater effort is considered to have a negative impact on the success of the project. We assume that the request for more effort tends to discourage the contribution of users.

Motivations are the different factors that induce users to contribute to a particular project. These are usually subjective and difficult to identify and measure.

In this study monetary incentives are not considered motivational factors because contributing to a CC project does not yield a direct economic return. Therefore, this study considers social factors as the main motivational factors (von Hippel 1988, Johnson 2002, Harhoff, Henkel et al. 2003, Lakhani and Wolf 2005, Lerner and Tirole 2005, von Hippel 2005, Valentinov 2007).

Usually, Nonprofit creators tend to be more successful in attracting contributions (Valentinov 2007). Indeed, they use social factors to compensate for lower salaries and incentives and to reward volunteering and donations (Stewart, Ammeter et al. 2006, Lambert 2010, Belleflamme, Lambert et al. 2013).

Producers who use open licenses and/or are prestigious tend to be more successful in attracting contributions from users because they use social factors to compensate for lower salaries and incentives and to reward volunteering and donations (Lerner and Tirole 2002, Lessig 2004).

In the following paragraph we describe the motivations used in the utility functions to define the behaviors of agents (users and producers).

3.2 The Utility Function

In this study the Cobb-Douglas (CD) utility function is used to simulate the behavior of agents:

$$U(x_1, \dots, x_n) = \prod_{i=1}^{n} x_i^{\alpha^i}$$

This utility function is used to evaluate the quality of a potential project and determine the users' potential contribution. Agents choose to contribute to projects hosted on the *prior-platform* on the basis of this utility function. Once projects have received a sufficient number of contributions they are removed from the *prior-platform* and published.

This model allows for an analysis of creative commons production in the context of an environment reminiscent of Kickstarter (Mollick 2013).

The x_i^{α} in the utility function are the different attributes of each project that agents take into account:

- x_1^{α} (effort) represents the propensity to participate in a project in light of the maturity of the project. The more mature the project, the less "effort" is needed. The value of x_1 depends on the *maturity* of the project. above, The value of the *maturity* of a project can be from 1 (minimum) to 10 (maximum). The value of x_1 is normalized to range from 1 (minimum) to 2 (maximum).
- x_2^{α} (prestige) represents the propensity to be attracted by prestigious projects. The value of x_2 depends on the characteristic of the producer—1 if the producer is not prestigious or 2 if the producer is prestigious.

- x_3^α (status) represents the propensity to be attracted by Nonprofit associations. The value of x_3 depends on the status of the producer, it is 1 if the producer is For-Profit and 2 if the producer is Nonprofit.
- x_4^α (dissemination) represents the propensity to be attracted by licenses that are open from a dissemination perspective. The value of x_4 depends on the openness in dissemination of the project license (see Table 2): 1 if minimum, 1.5 if medium, and 2 if maximum.
- x_5^α (control) represents the propensity to be attracted by licenses that provide an opportunity to co-produce and modify the original work. The value of x_5 depends on the openness in production of the project license (see Table 2): 1 if minimum, 1.5 if medium and 2 if maximum.

We add another attribute, x_6^α (skip), to accommodate the possibility that the agent decides to not participate in a project. A value of x_6 equal to 2 represents the utility to the agent of the blank idea (do nothing); otherwise it is equal to 1.

α^i represents the sum of α in each utility function. Each α represents the weight of each factor in the different utility functions.

The CD utility function does not strictly require that the weights assigned to the factors sum to 1. However, it is useful to normalize the utility function by constraining our weights to sum 1 both for algebraic convenience (Varian 2000) and to allow inter-agent comparisons (Elster and Roemer 1991, Brown and Robinson 2006).

3.3 Model Description

As mentioned above, our model universe consists of agents and projects. We assume that creators of new projects act as producers and contributors act as users. Moreover, agents can also decide to do nothing.

In this model each contribution to a project is represented by a one-unit decrease in the contribution required to complete the project (*maturity*).

The description of the actions in each step of the model is as follows:

- "Action 0": the model is initialized and *n* agents are created. The values of the *characteristics* (commercial, prestigious, maturity, and license) that each agent transfers to each project are randomly assigned.
- "Action 1": an agent is chosen randomly who "imagines" 7 different projects, one for each CC license, plus 1 Public Domain (PD) and plus 1 blank idea. The blank idea indicates the non-creation of a project. The licenses are organized according to the degree of openness (*minimum*, *medium* and *maximum)* in production as well as in dissemination (Gambardella 2011) (see Table 2).
- "Action 2": these "imagined" projects are stored in a sort of "prior-platform." The prior-platform contains the 8 "imagined" projects and eventually the projects already selected by agents in previous steps.[5]
- "Action 3": the selected agent finds which project in the *prior-platform* maximizes his or her utility function. This agent then contributes effort to the "selected project." An agent who selects one of the imagined projects acts as a

[5] There are 0 prior projects at the first step.

"producer," while one who selects a project already chosen in a previous step acts as a "user." If the agent selects the "blank idea," the agent does nothing and no project is created.

- "Action 4": if the selected project is the blank idea, all imagined projects are dropped and we go back to "Action 1." If the selected project is one of the 7 imagined, this project is kept in the *prior-platform* with a number of contributors equal to 1 and the other 6 imagined projects are dropped. If the selected project is one of the projects already presented in the *prior-platform*,[6] a contribution equal to 1 is added to this project and all the imagined projects are dropped.
- "Action 5": the *maturity* of projects is checked and if the project is complete it acquires the status of "completed project." A project is completed when the number of agents having contributed equals the value of "maturity," randomly assigned at "Action 0." The complete project is removed from the *prior-platform* and published on the *visible-platform*.
- "Action 6": we go back to "Action 1" to start another step until the chosen number of steps is completed.

We notice that at the first step there are only 7 projects plus a blank idea in the *prior-platform*. The 7 projects in the *prior-platform* are the selected agent's 7 "imagined projects."

At the first step, the selected agent can only choose from among his or her 7 imagined projects or do nothing (the blank idea). If the agent decides to do nothing the *prior-platform* will be empty, otherwise we will have 1 project stored on it.

[6] This means that it is a project previously selected from an agent's 7 imagined projects.

At the second step, the number of "evaluable projects" consists of the project that may already be stored in the *prior-platform* plus the 7 imagined projects of the new randomly chosen agent.[7] Then this agent chooses the utility-maximizing project.

If the agent chooses to do nothing, all imagined projects are dropped and only the previously chosen project is kept in the *prior-platform*. In this case the number of contributors to this project does not change.

If the agent chooses to contribute to a project already stored in the *prior-platform,* all imagined projects are dropped and only the previously chosen project is kept in the *prior-platform*. In this case, the number of contributors increases by 1.

If the agent chooses to contribute to one of his or her 7 imagined projects, the other imagined projects are dropped and this chosen project is stored in the *prior-platform* together with previously chosen projects. In this case the number of contributors to the newly chosen project increases by 1 and the number of contributors to previously chosen projects does not change.

This process continues until the number of steps is completed. A project is completed when the number of contributors is equal to the value of *maturity* required to complete the project. In this case, the project succeeds and is published.

The model is repeated for each year. In our empirical database we have 7 years to simulate (2003 to 2009).

Each project contains the information about its own *characteristics*. Agents' characteristics are constant over the years; this is important for tracking the different

[7] We notice that previously chosen agents can be selected again.

projects. Indeed, the characteristics of the agents represent the characteristics of the projects.

The *attributes* of agents change over the years because we need to have all combinations of users' motivations. The *attributes* represent the motivations of users, they are the arguments of each agent's utility function. However we do not track them because we are not interested in changes in contributors' motivations.

In other words, projects started by the same agent have the same characteristics, but agents that contribute in different years can be either different agents or the same agent, but with different motivations. In this way we can track the attractiveness of characteristics of projects.

On the basis of the number of contributions, participants can estimate how much effort is required to complete the project. Less effort is, of course, preferred.

The attributes in the utility function of each agent indicate the weight assigned to the characteristics of each project.

As an example, at the first step of the model simulation Agent_64 is randomly chosen. He imagines 7 projects plus a blank idea and decides which of the 8 projects maximizes his utility function (7 imagined projects plus a blank idea). Then Agent_64 solves his different utility function for the 7 different licenses and the blank idea. For example, the utility function of Agent_64 and project under CC BY-NC is:

$$U_{64_{by-nc}} = effort^{\alpha_1}\ status^{\alpha_2}\ control^{\alpha_3}\ dissemination^{\alpha_4}\ prestige^{\alpha_5}\ skip^{\alpha_6}$$

Then we use the value of the characteristics of Agent_64 (*effort*, *status*, *prestige*), the degree of openness of the license (*control*, *dissemination*), and the blank idea (*skip*) to assign values to the attributes in the utility function. Finally, we assign weights to his *attributes* in the utility function $(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6)$ reflecting the motivations of the agent.[8] The value of *effort* is normalized to reflect the maximum possible value of the required effort. We let the maximum possible value be 10. Then the necessary effort is equal to 2, and *effort* = 1 + (2/10) = 1.2.

The values of *status* and *prestige* are equal to 2 if the agent is *Nonprofit* and *prestigious*, respectively, 1 if not.

The values of *control* and *dissemination* are equal to 2 (maximum) or 1.5 (medium) or 1 (minimum) according to the degree of openness in *production* and *dissemination* respectively (see Table 2):

$$U_{64_{by-nc}} = 1.2^{0.07}\ 2^{0.30}\ 2^{0.23}\ 1^{0.31}\ 2^{0.05}\ 1^{0.04}$$

This means that this agent is more attracted by the Nonprofit status ($\alpha_2 = 0.30$) and a license that is open in production ($\alpha_3 = 0.23$) and dissemination ($\alpha_4 = 0.31$). The necessary effort to complete the project ($\alpha_1 = 0.07$), the prestige of the project ($\alpha_5 = 0.05$), and doing nothing ($\alpha_6 = 0.04$) are not very important to this agent.

In case of a blank idea all *attributes* except *skip* are equal to 1. *Skip* is equal to 2. In this case we take into account only the weight of doing nothing:

$$U_{64_{blank}} = 1^{0.07}\ 1^{0.30}\ 1^{0.23}\ 1^{0.31}\ 1^{0.05}\ 2^{0.04}$$

---

[8] For this example values calculated with the results of the calibration procedure of paragraph 4 were used.

Once Agent_64 solves his utility function for all projects, he keeps the one that yields the highest utility. Then this project is stored in the *prior-platform* and the other projects are dropped. In this case Agent_64 acts as producer.

In the second step, Agent_94 is chosen at random. She imagines the 8 projects (7 projects for the CC licenses and the blank idea) and solves her utility function for the 8 projects plus the project already stored in the *prior-platform*. It happens that the highest utility of Agent_94 results from using the characteristics of the project created by Agent_64 and already stored in the *prior-platform*. Thus, Agent_94 contributes to the project started by Agent_64. All 8 projects imagined by Agent_94 are dropped. In this case Agent_94 acts as contributor.

Each contribution to the project started by Agent_64 decreases the effort needed to complete the project. It starts with *maturity* equal to 4. After the first contribution by Agent_64 and then that by Agent_94, the necessary effort required to complete the project is equal to 2. When the necessary effort is equal to 0 the project is completed and published.

The model continues selecting other agents randomly at each step until the number of steps is completed.

We arbitrarily set the number of agents at 100, since this number is more than enough to generate different combinations of characteristics of agents. These characteristics are randomly assigned. The weights of *attributes* in utility functions and the number of steps are calibrated so that the model replicates the results of the empirical database.

4 Calibrations

In order to properly investigate and validate our simulation we use a joint analysis methodology (Kennedy, Xiang et al. 2005) that has already been used for ABM including individuals (such as producers) (Garcia, Rummel et al. 2007).

To validate our ABM we compare the simulated results from the *prior-platform* to the behavior empirically observed in the real *visible-platform*. The empirical database was created from data in the IA (Gambardella 2011). For this study we only selected videos licensed under CC or PD and created by Nonprofit and For-Profit producers from the original database.

Previous research has shown that CC licenses impact the behavior of producers and users (Lerner and Tirole 2002, Lerner and Tirole 2005, von Hippel 2005, Colazo and Fang 2009). Nonprofit producers and open licenses are more likely to attract contributions from users (Valentinov 2007, Gambardella 2011).

The ABM presents different parameters to which values have to be assigned. Since a random search of the parameters is not practical and will not cover all possible combinations, an important challenge of this study was calibration of the parameters. Using an "iterated racing procedure" called *iRace* (López-Ibánez, Dubois-Lacoste et al. 2011) we obtained different sets of parameters to calibrate our model in order to replicate the empirical results. The output of the different scenarios offers the opportunity to capture decision-making behavior and reveals strategies to attract contributions from users (Fagiolo and Roventini 2012).

Iterated racing is an automatic configuration method that consists of three steps: (1) sampling new configurations according to a particular distribution, (2) selecting the best configurations from the newly sampled ones by means of racing, and (3) updating the sample distribution in order to bias the sampling towards the best configurations (López-Ibánez, Dubois-Lacoste et al. 2011).

First, a target is defined. In our case we use as the target the number of projects published each year by Nonprofit and For-Profit producers in our empirical database.

(Table 3 about here)

To run the calibration some parameters have to be defined:

- the number of agents;
- the range of steps;
- the computational budget .

As mentioned above we decided to use 100 agents. We also tested the model with 10 and 1000 agents and the results were similar: The model does not appear to exhibit scale effects. Once the agents are created, the model randomly assigns values to their characteristics. These values do not change during the instances of iRace.

The range of steps, s, indicates the minimum and maximum number of steps of each instance of the model. We define $s = [10, 2000]$ the range of steps. We consider 2000 to be sufficient as the maximum number of steps because in previous tests the number of steps necessary to reach the target was less than 500.

The computational budget determines the maximum number of experiments allowed to reach the target. Each experiment tests a random configuration of α_n. In each experiment, every configuration has different vectors of α_n. We define 50,000 as the maximum computational budget because previous tests with a lower budget did not produce results.

With this procedure we selected the value of each characteristic of the different projects and the value of α for each attribute in the different utility functions for each year.

iRace allows for different scenarios, but we selected only those that were able to reach the targets in each year, as mentioned above.

Each scenario contains the characteristics of agents, a weight for agents' attributes and the necessary number of steps.

(Table 4 about here)

The characteristics of the agents are the same in each scenario. Once agents have transferred their characteristics to the project, these characteristics do not change over time or with the scenarios. The weight on attributes, which represent the motivations of agents, changes in each scenario.

Moreover, the procedure provides the probability, p, of choosing the best project (the project that maximizes the utility function of the agent).

$$p = \frac{m^r}{m^r + M^r}$$

m represents a project that does not maximize the utility function and M represents the project that does. The value of r represents the sensitivity of agents to the differences among projects. The higher the value of r, the greater the probability that agents choose exactly those projects that maximize their utility functions. For example, if r

equals 1 and Agent_1's project yields utility equal to 1.8 and Agent_2's project yields utility equal to 1.9, the former has a 49% probability of being chosen; if *r* is equal to 100 this probability decreases to 0.4%. In each scenario the value of *r* is the same, representing the stickiness of information (see Table 3). The stickiness of information indicates the incremental expenditures required to obtain a unit of information (von Hippel 1998). In our model, this represents the possibility that agents know all the characteristics of potential projects and then choose the best one. We observe a constant *r* from 2005 to 2009; indicating that the stickiness of information remained constant during those years. We observe that 2004 has a high *r* compared to other years, meaning that during this year it was possible for agents to select the best project. We interpret this result as the consequence of a decline in the level of stickiness of information.

Each scenario selected replicates the results of each year observable in the empirical database.

5 Results

Our model respects the fundamental ingredients of a "canonical" ABM (Fagiolo and Roventini 2012). The model is able to reproduce the key characteristics observed in the real platform mentioned above. Moreover, the model is able to open the "black box" and show the hidden dynamics that cause the results that are visible on the real platform. What we observe on the real platform is sort of a result of competition among projects. Only projects that are able to collect enough contributions survive and can be observed. We cannot observe uncompleted projects and we cannot observe the motivations that incite users to contribute to a project. The model is able to provide an explanation for these underlying dynamics.

An important goal of our ABM is to mimic the hidden behavior of producers. Empirical data only reveals the characteristics of producers who survive (who publish their works). Our ABM is capable of modeling the different attributes of each agent's utility function.

Given an initial number of agents, the model provides different scenarios for each year. Each scenario contains the different attributes of utility functions and the characteristics of projects for each agent. Each scenario is able to replicate the real platform.

We test our ABM using 100 agents, from 10 to 2000 steps, and 50,000 computational budgets. After calibration we obtain different scenarios for each year.

In each scenario, the characteristics of agents remain the same while the weights on the attributes change. This means that potential producers are the same in each scenario, but the motivations of contributors change.

Stochastic components are included in the model. To be sure of the results, 100 runs are performed for each scenario. The average results of the runs for each scenario are compared to the empirical data.

As in history-friendly models (Franco Malerba and Winter 2001), we compare the stylized facts of the real phenomenon with the results of our model.

It clearly emerges that our model is able to mimic the general results we have on the real platform (Figure 1). With the parameters and characteristics of agents as inputs, the model's output is consistent with the empirical data. Indeed, the number of projects that succeed each year is consistent with the number of works we observe on the real platform.

Moreover, the model is able to distinguish between, and simulate, the behavior of both Nonprofit and For-Profit agents.

In the case of For-Profit production (Figure 2) we easily observe an increase in production starting in 2005 and ending in 2007, followed by a downturn. In the case of Nonprofit production (Figure 3) we easily observe that production increases over the entire period. In both the For-Profit and the Nonprofit cases, the simulated plot is close enough to the plot of empirical data.

Our real data shows an increase in CC works in the IA over time, particularly during the period 2005–2007, and particularly produced by For-Profit producers. We interpret this data as the consequence of the fact that, as shown by a Google Trends[9] plot (Figure 4), from 2005 to 2007 Creative Commons licenses became much more well-known. Indeed, at the end of 2004 the popular magazine Wired, in collaboration with the Creative Commons organization and sixteen musicians, assembled the first major compilation of music that was free to sample and share under CC.[10] In 2006, Microsoft and the Creative Commons organization released a tool to license works under CC. In 2007, Wikipedia contents became licensed under CC.[11]

---

[9] Google Trends is a tool that shows how often a particular term is searched on Google, we tested the term: Creative Commons from the beginning of 2004 to the end of 2009. It was not possible to test from 2003 because Google Trends starts from 2004

[10] http://creativecommons.org/wired/

[11] http://tech.slashdot.org/story/07/12/01/2032252/wikipedia-to-be-licensed-under-creative-commons

Notwithstanding standard property rights theory, according to which the attenuation of property rights eradicates the motivation to produce, we observe an increase in the use of licenses (CC and FLOSS) that attenuate property rights (Lessig 2004, Boldrin and Levine 2008). This happens because alternative incentives such as *non-monetary* (reputation, career concerns, peer recognition, and sharing innovation) and *intrinsic* (activity itself, ego gratification, and need) motivations (von Hippel 1988, Johnson 2002, Lerner and Tirole 2002, Harhoff, Henkel et al. 2003, Lakhani and Wolf 2005, von Hippel 2005, Valentinov 2007) are able to attract contributions from users. Non-monetary and intrinsic motivations are represented as attributes in the utility functions of our agents. Indeed a *prestigious* project is more able to attract users motivated by *non-monetary* motivations; the necessary *effort* and the propensity to do nothing (*skip*) impact on the people motivated by *intrinsic* motivations. Our model is able to replicate such a selection process leading to the success of some projects.

Moreover, our model mimics the effect that *For-Profit* producers are more attracted to CC licenses because they need to be more attractive than *Nonprofits* (Valentinov 2007). Thus, we are not surprised to find more projects under CC created by *For-Profit* agents than by *Nonprofit* ones.

To better investigate the impact that variables representing the characteristics of a project (*commercial, prestige, effort*) have on its *success* (dependent variable) a linear regression was estimated. The output of the model was used in the regression.

The *p*-value indicates that our model is statistically significant. The Brant test confirmed that our model is statistically significant. All results are significant ($p < 0,001$).

The results (see Table 1) are consistent with our expectations. The regression results show that, despite the classical property rights approach in which for-profit producers need a strong copyright to be motivated to produce, an alternative way is possible. The use of CC licenses reduces copyright power. The regression results confirm that a *for-profit* status has a negative impact on the success of a project and consequently *for-profit* agents need to renounce part of the copyright by using more open CC licenses if they are to attract contributions from users and succeed.

Our model is able to show variables that we cannot observe in the empirical database. The variable *commercial* in the empirical database can only be observed for projects that succeed—which are the ones visible on the online platform. In our model we also have characteristics of the producers who fail.

In the empirical database there is no information on the *prestige* of the producers and the *necessary level of effort*.

The *prestige* of the producer is not easy to observe. Indeed, a producer might be famous within a certain community of users, but unknown elsewhere. For example, an artist can be well-known in a certain circle, but totally unknown on the Internet. This artist is able to use his or her celebrity to collect funds to produce a video, but it is impossible for us to empirically observe this dynamic. Our model is able to mimic the *prestige* of the producer in its community of users.

The *necessary effort* required by the producer to complete the project and succeed is information we cannot observe in the empirical data. Though some objective data can be collected (such as costs of the project), this is only possible for projects that succeed—we have no information on projects that fail. Moreover, from the point of view of contributors, the effort associated with each contribution is subjective. Our model is able to mimic the effort required from producers of projects that succeed as well as of those that fail. Moreover, our model mimics users' propensity to contribute from their subjective point of view.

8 Conclusions

Collecting contributions from users and leading the project to success is a challenge. It is important to understand the conditions that contribute to project success for the management of CC projects. To explore the conditions that contribute to the success of a CC project, this study uses an ABM of CC production. The model is built on assumptions regarding users' motivations, which have already been analyzed in the literature, and calibrated using empirical data from the IA. In the case of projects under open licenses such as CC licenses, users prefer to contribute to prestigious projects because they receive non-monetary compensation motivated by things like peer-recognition, career concerns, reputation, etc. Our model accounts for these motivations.

The model reveals that in CC production the success of a project depends on its own characteristics and on its capacity to attract contributions from users. The status and prestige of the producer and the effort required by the project are important factors. These results are consistent with the literature on open licenses.

The model helps to observe the impact of unobservable variables on the empirical database. Indeed, we can only collect data on projects that succeeded from the empirical database, and we do not have any data on projects that failed. Moreover, the subjective perception of the effort and benefits associated with contributing is not observable in the empirical database.

Our model is able to mimic these unobservable data and the behavior of producers and contributors. As a result, we can observe how the characteristics of projects attract contributions from users and, thus, how they succeed.

This study contributes to alleviating the shortage of literature on CC production. To our knowledge, this is the first model of CC evolution that includes the role of users' utility and the projects' characteristics in determining the success of projects.

The main challenge posed by this model is calibrating it with empirical data. To accomplish this, we used an iterated procedure that tests different combinations of values and generates scenarios that replicate the results in the empirical database. Our model is able to replicate the pattern observed in the empirical data.

The model examines results drawn from a real selection of projects. Different scenarios are run to generate the target values. Each scenario contains the characteristics of projects that succeed and that fail and the weight of attributes in the utility functions of users, who may contribute or not. The utility functions represent agents' motivations to create and/or contribute to a project.

Using this data as the input, the model is able to replicate the behavior of users and producers. Then we can observe which projects succeed or fail. We can also observe the contributions of users. The model is able to mimic the selection process and generate the total amount of production observed in the empirical data. It is also able to replicate the increase and decrease in production over time and to distinguish and replicate production by For-Profit and Nonprofit agents. Moreover, the model is able to replicate the impact the characteristics of a project have on its success.

In conclusion, the model is useful for acquiring a better understanding of the conditions necessary for the success of CC projects.

Further implementations of the model are necessary to better analyze the characteristics that allow CC projects to succeed.

Such implementations of the model could focus on three directions.

A first implementation could examine the different variables in agents' utility functions. This would help to better understand the behavior of users: when and why they do or don't contribute to projects.

A second possible implementation could be a representation of the degree of openness of licenses. This may help to better describe which other requirements are necessary for a project to succeed.

A third possible implementation would be the calibration on other databases of CC works (music, texts, pictures, software). This may help to better understand CC production in general.

These implementations would make it possible to have a more complete representation of the conditions determining the success of CC projects in real life.

References

Belleflamme, P., T. Lambert and A. Schwienbacher (2013). "Crowdfunding: Tapping the Right Crowd." Journal of Business Venturing, Forthcoming. CORE Discussion Paper No. 2011/32 1-39.

Boldrin, M. and D. Levine (2008). Against Intellectual Monopoly. Cambridge University Press**:** 1-325.

Brown, D. and D. Robinson (2006). Effects of heterogeneity in residential preferences on an agent-based model of urban sprawl. Ecology and society.

Colazo, J. and Y. Fang (2009). Impact of license choice on Open Source Software development activity. Journal of the American Society for Information Science and Technology. **60:** 997-1011.

Dalle, J. M. and P. A. David (2005). SimCode: Agent-based Simulation Modelling of Open-Source Software Development. The Economics of the Internet.

David, P. (1985). Clio and the Economics of QWERTY. The American Economic Review.

Elster, J. and J. Roemer (1991). Interpersonal comparisons of well-being. Cambridge University Press.

Fagiolo, G. and A. Roventini (2012). Macroeconomic policy in dsge and agent-based models. Revue de l'OFCE, OFCE**:** 67-116.

Franco Malerba, R. N. L. O. and S. Winter (2001). History-Friendly models: An overview of the case of the Computer Industry

, JASSS. **4**.

Gambardella, M. (2011). The Scope of Open Licenses in Cultural Contents Production and Distribution. EconomiX Working Papers, University of Paris West - Nanterre la Défense, EconomiX. **26**.

Garcia, R., P. Rummel and J. Hauser (2007). Validating agent-based marketing models through conjoint analysis. Journal of Business Research.

Harhoff, D., J. Henkel and E. v. Hippel (2003). Profiting from voluntary information spillovers: how users benefit by freely revealing their innovations. Research Policy. **32:** 1753-1769.

Johnson, J. (2002). Open source software: Private provision of a public good. Journal of Economics & Management Strategy.

Kennedy, R., X. Xiang, G. Madey and T. Cosimano (2005). Verification and validation of scientific and economic models. Proc. Agent.

Lakhani, K. and R. Wolf (2005). Why hackers do what they do: Understanding motivation and effort in free/open source software projects. Perspectives on free and open source software**:** 3–22.

Lambert, T. (2010). An empirical analysis of crowdfunding. Social Science Research Network ….

Lerner, J. and J. Tirole (2002). Some simple economics of open source. The journal of industrial economics. **50:** 197-234.

Lerner, J. and J. Tirole (2005). The scope of open source licensing. Journal of Law, Economics, and Organization, Oxford Univ Press. **21:** 20-56.

Lessig, L. (2004). Free culture: How big media uses technology and the law to lock down culture and control creativity. books.google.com.

Lessig, L. (2004). Free culture: How big media uses technology and the law to lock down culture and control creativity. The Penguin Press.

López-Ibánez, M., J. Dubois-Lacoste, T. Stützle and M. Birattari (2011). The irace package: Iterated racing for automatic algorithm configuration, Tech. Rep.

TR/IRIDIA/2011-004, Université Libre de Bruxelles, IRIDIA, Av FD Roosevelt 50, CP 194/6 1050 Bruxelles, Belgium, http://iridia. ulb. ac. be/irace.

Malerba, F., R. Nelson, L. Orsenigo and S. Winter (2006). A History-Friendly Model of the Co-Evolutionof the Computer and Semiconductoris Industries: Capabilities and Technical Change as Determinants of the Vertical Scope of Firms in Related Industries. unibg.it. **working paper**.

Mollick, E. (2013). The dynamics of crowdfunding: An exploratory study. Journal of Business Venturing.

Mustonen, M. (2010). Economics of Creative Commons.

Radtke, N. P., M. A. Janssen and J. S. Collofello (2009). What Makes Free/Libre Open source software (FLOss) Projects successful?

Stewart, K., A. Ammeter and L. Maruping (2006). Impacts of license choice and organizational sponsorship on success in open source software development projects. Information Systems Research. **17:** 126-144.

Tesfatsion, L. and K. Judd (2006). Handbook of Computational Economics. Handbook of computational economics, Elsevier. **2:** 831-880.

Valentinov, V. (2007). The Property Rights Approach to Nonprofit Organization: The Role of Intrinsic Motivation. Public Organization Review. **7:** 41-55.

Varian, H. (2000). Variants in economic theory: selected works of Hal R. Varian. Economists of the twentieth century. Cheltenham, UK ; Northampton, MA : Edward Elgar.

von Hippel, E. (1988). The sources of innovation. Oxford University Press, New York**:** 218.

von Hippel, E. (1998). Economics of Product Development by Users: The Impact of "Sticky" Local Information. Management Science. **44:** 629-644.

von Hippel, E. (2005). Democratizing Innovation. The MIT Press: 224.

Appendix

Figure 1

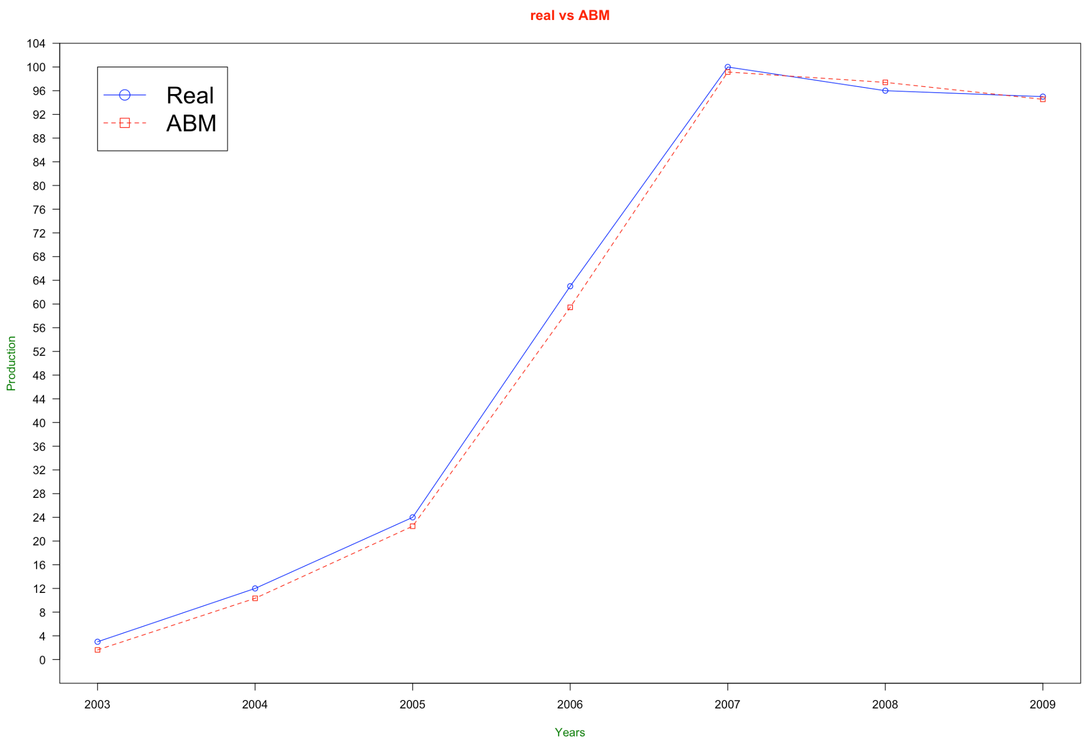
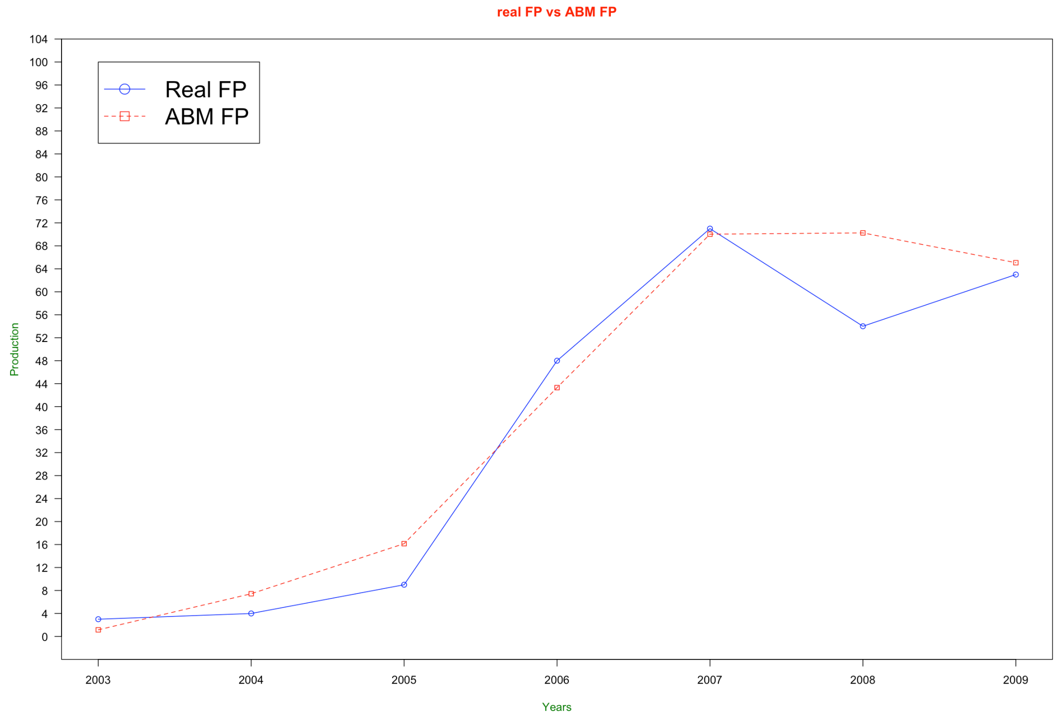
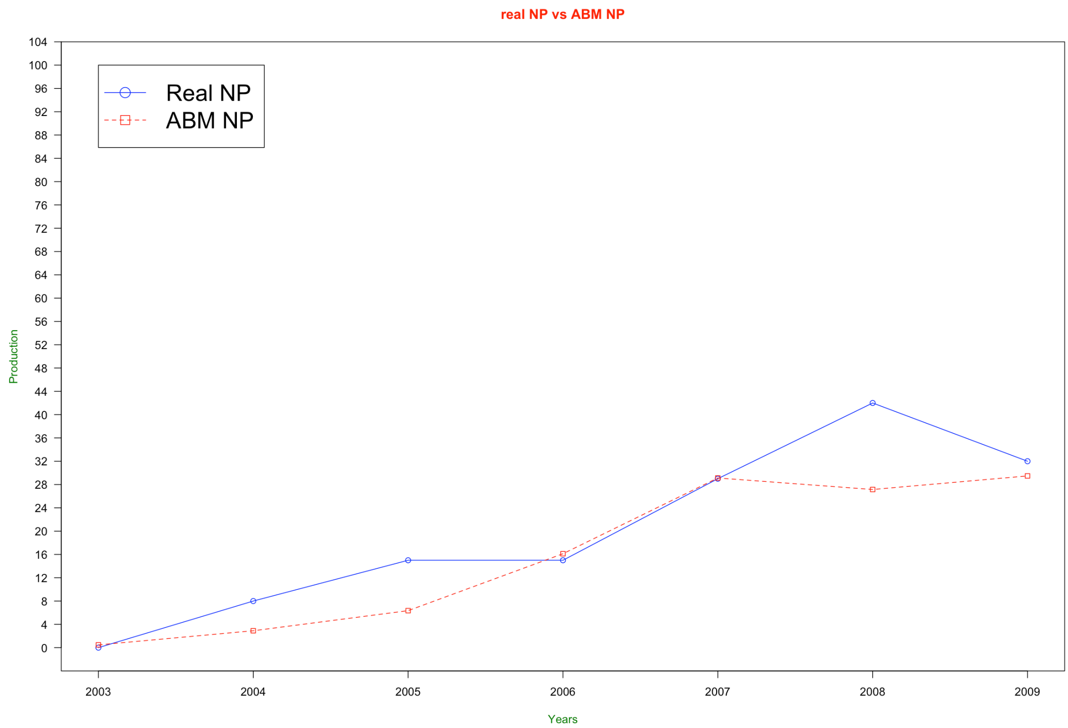


Figure 2



real FP vs ABM FP
Real FP
ABM FP
Production
Years

Figure 3



real NP vs ABM NP
Real NP
ABM NP
Production
Years

Figure 4

Table 1

Table 1: Impact on Success

Variable	**Coefficient** (Std. Err.)
commercial	-0.311** (0.046)
famous	0.494** (0.047)
difficulty	-0.063** (0.008)
Intercept	0.808** (0.055)
N	700
R^2	0.251
$F_{(3,696)}$	77.739

Significance levels : † : 10% $*$: 5% $**$: 1%

Table 2

Degree of openness	**Production**	**Diffusion**
Maximum	PD CC by	PD CC by
Medium	CC by-sa CC by-nc CC by-nc-sa	CC by-sa CC by-nd
Minimum	CC by-nd CC by-nd-nc	CC by-nc CC by-nc-sa CC by-nc-nd

Table 3

	Targets						
	2003	2004	2005	2006	2007	2008	2009
For-profit	3	4	9	48	71	54	63
Nonprofit	0	8	15	15	29	42	32

Table 4

	2003	2004	2005	2006	2007	2008	2009
n. of scenarios	4	3	2	5	4	2	5

Table 5

	r by years						
Scenario	**2003**	**2004**	**2005**	**2006**	**2007**	**2008**	**2009**
1	21	64	10	11	10	10	10
2	11	97	12	10	10	10	10
3	70	92	-	10	11	-	10
4	16	79	-	11	11	-	10
5	-	-	-	12	-	-	11